

Scraping y buenas prácticas de programación web



Grado en Ingeniería Informática

Trabajo Fin de Grado

Ana María Molina Primicia

Alfredo Pina Calafi

Pamplona, 1 de Julio de 2016

RESUMEN

Este proyecto trata sobre la realización de una máquina cuyo cometido es recoger datos de páginas web y visualizarlos en otra web añadiendo estadísticas sobre deporte base local, en concreto fútbol.

El trabajo constará de tres fases diferenciadas:

1. Obtención de una serie de datos de unas páginas web concretas, así como de algún archivo Excel, que se almacenarán en una base de datos. Esto se realizará mediante la técnica llamada scraping.
2. Visualización de los datos en una web en la que se les dará un uso diferente realizando estadísticas y gráficos.
3. Realización de pruebas de usabilidad a la web realizada, así como el análisis de la accesibilidad.

Tras realizar las dos primeras fases obtendremos una página web capaz de administrarse ella misma.

Todo esto incluirá las tareas de análisis y diseño funcional de las dos etapas, toma de decisión de cuál será la mejor tecnología a usar en el lado servidor, así como el aprendizaje de nuevas técnicas de visualización de datos.

PALABRAS CLAVE

Scraping, web, análisis web, usabilidad, accesibilidad, visualización de datos

ÍNDICE

| | |
|---|----|
| RESUMEN | 2 |
| PALABRAS CLAVE | 2 |
| CAPITULO I INTRODUCCIÓN | 5 |
| 1. INTRODUCCIÓN | 6 |
| 1.1 OBJETIVO | 6 |
| 1.2 CONTEXTO TECNOLÓGICO | 6 |
| 1.2.1 SCRAPING WEB | 7 |
| 1.2.2 FRAMEWORK DJANGO | 7 |
| 1.2.3 LIBRERÍA D3.JS | 8 |
| CAPITULO II REQUISITOS Y ANÁLISIS FUNCIONAL | 9 |
| 2. REQUISITOS Y ANÁLISIS FUNCIONAL | 10 |
| 2.1 REQUISITOS | 10 |
| 2.1.1 REQUISITOS DE LA APLICACIÓN DE SCRAPEO | 10 |
| 2.1.2 REQUISITOS DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS | 11 |
| 2.2 ANÁLISIS FUNCIONAL | 12 |
| 2.2.1 DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN DE SCRAPEO | 12 |
| 2.2.2 DIAGRAMA DE ACTIVIDAD DE LA APLICACIÓN DE SCRAPEO | 19 |
| 2.2.3 DIAGRAMA DE SECUENCIA DE LA APLICACIÓN DE SCRAPEO | 20 |
| 2.2.4 DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS | 21 |
| 2.2.5 DIAGRAMA DE ACTIVIDAD DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS | 28 |
| 2.2.6 DIAGRAMA DE SECUENCIA DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS | 29 |
| | 29 |
| 2.3 ANÁLISIS BASES DE DATOS | 30 |
| 2.3.1 BBDD FUTNAVARRA | 30 |
| 2.3.2 BBDD INTERMUNICIPAL | 35 |
| 2.3.3 BBDD FÚTBOL TXIKI | 39 |
| CAPITULO III | 43 |
| MANUAL DE USUARIO | 43 |
| 3. MANUAL DE USUARIO | 44 |
| 3.1. MANUAL DE USUARIO APLICACIÓN DE ESCRAPEO | 44 |

| | |
|--|----|
| 3.2. MANUAL DE USO DE LA WEB DE DEPORTE BASE..... | 48 |
| CAPITULO IV DESPLIEGUE..... | 57 |
| 4. DESPLIEGUE..... | 58 |
| CAPITULO V ACCESIBILIDAD..... | 62 |
| 5. ACCESIBILIDAD..... | 63 |
| CAPITULO VI PRUEBAS DE USABILIDAD | 65 |
| 6. PRUEBAS DE USABILIDAD | 66 |
| CAPITULO VII PROBLEMAS Y SOLUCIONES | 67 |
| 7. PROBLEMAS ENCONTRADOS Y SOLUCIONES PROPUESTAS | 68 |
| 7.1 ESCRAPEO FUTNAVARRA | 68 |
| 7.2 LECTURA DE ARCHIVO EXCEL..... | 68 |
| 7.3 GRÁFICAS DE LA WEB DE VISUALIZACIÓN..... | 69 |
| 7.4 ERROR EN EL DESPLIEGUE..... | 69 |
| 7.5 IMPLEMENTACIÓN INCOHERENTE FUTNAVARRA..... | 69 |
| CAPITULO VIII CONCLUSIONES..... | 70 |
| 8. CONCLUSIONES Y MEJORAS FUTURAS | 71 |
| ANEXOS | 73 |
| ANEXO I: FRAMEWORK DJANGO | 74 |
| ANEXO II: CUESTIONARIO SUS..... | 77 |
| ANEXO III: PRUEBAS DE USABILIDAD | 78 |
| ANEXO IV: PLANTILLA EXCEL | 79 |
| BIBLIOGRAFÍA | 81 |

CAPITULO I INTRODUCCIÓN

1. INTRODUCCIÓN

1.1 OBJETIVO

El objetivo principal es la realización de una página web usable y accesible para poder mostrar datos referidos al deporte base en Navarra, en concreto el fútbol, cogiendo datos automáticamente de otras páginas web. Es decir, tener una página web que se alimente automáticamente de datos de otras páginas web ahorrando tiempo de administración. También quitar la necesidad de tener un administrador introduciendo datos ya que cualquier usuario con un simple clic podrá realizar esta tarea.

Para conseguir este objetivo hay que realizar dos trabajos diferenciados:

1. Realización de una máquina que recoja datos de páginas web de forma automática usando la técnica llamada 'scraping'. Estos datos se almacenarán en una base de datos para su posterior uso.
2. Diseñar y crear una página web usable y accesible para poder mostrar estos datos. Los datos se visualizarán en forma de estadísticas y gráficos, además de la forma en tablas habitual.

Con esto queda claro que se trata de la realización de dos aplicaciones web. Una que nos de la opción de escrapear unas páginas web para recoger los datos necesarios sobre el deporte base local. Y otra para visualizar estos datos que habrán sido almacenados en una base de datos.

Tras la realización de las dos aplicaciones se hará un test de usabilidad y accesibilidad a la web de visualización de datos para verificar que cumple los estándares.

1.2 CONTEXTO TECNOLÓGICO

Para la realización de este proyecto se han utilizado varias tecnologías que deben ser mencionadas:

- Técnica de 'scraping'
- Framework Django
- Librería d3.js

1.2.1 SCRAPING WEB

En términos generales, el uso del scraping web trata de simular el comportamiento de un humano navegando por la web, para recoger datos del HTML (sin estructura) y poder darles una estructura para su posterior uso. En los últimos años, se ha convertido en una técnica muy utilizada dentro del sector del posicionamiento web ya que permite recoger gran cantidad de datos de forma automática para poder usarlos posteriormente.

En este proyecto se ha decidido utilizar la librería 'Beautiful Soup' [1] de Python. Esta librería nos facilita el proceso de scraping ya que sirve para parsear documentos HTML. Su objetivo es construir un árbol de etiquetas HTML en la que es muy fácil extraer la información deseada. Hay que mencionar que para que sea fácil extraer la información hay que realizar un análisis previo del HTML a escrapear para saber dónde está la información que se necesita.

Hay que tener en cuenta que a veces es difícil escrapear la información deseada dada la mala implementación de las páginas web. Si todas las etiquetas HTML contaran con su identificador 'id' este proceso sería mucho más fácil y eficiente.

La elección de esta tecnología ha venido condicionada por el uso del lenguaje Python en el lado del servidor, ya que se ha decidido utilizar el framework Django que pertenece a Python. Este framework se detalla a continuación.

1.2.2 FRAMEWORK DJANGO

Primero es conveniente explicar que es un framework. Un framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Trata de facilitar las actividades comunes a realizar durante el desarrollo web. Prácticamente todos los frameworks basan su estructura en el patrón MVC o MVP (Modelo Vista Controlador). Esto significa que separa el desarrollo en 3 componentes que evolucionan de forma independiente. Estas componentes son el modelo de datos (Modelo), la interfaz gráfica (Vista), y la gestión de las entradas del usuario (Controlador).

Django es un framework de Python de alto nivel que pretende ahorrar tiempo a los desarrolladores produciendo aplicaciones fáciles de mantener y que rindan bajo mucha carga. Algunas de sus ventajas son que fomenta el desarrollo de código limpio y ágil al promover buenas prácticas de programación web, desarrollo rápido, amigable con el usuario, etc. Como es un framework se basa en el modelo MVC usando 3 scripts de Python:

1. models.py que contiene el modelo de datos.
2. views.py que contiene la lógica de la página, el en cargado de construir la interfaz de usuario, lo que denominaríamos vista.
3. urls.py que contiene las especificaciones para saber a qué vista hay que llamar según el patrón URL, lo que denominaríamos controlador.

El funcionamiento es el típico de este modelo. El usuario cuando está en un sitio web visualiza un interfaz de usuario y realiza una acción cualquiera. Esta acción internamente manda una petición al servidor mediante una url que es interpretada por el controlador. El controlador, según la url recibida, llama a una función definida en `views.py` que mediante comunicación con el modelo recoge los datos requeridos y renderiza el HTML que será lo que finalmente visualizará el usuario.

Se ha decidido utilizar este framework para innovar un poco y no utilizar el lenguaje PHP en el lado del servidor que es lo más habitual. Además, nos facilita mucho la realización de la programación en cuanto a consultas de bases de datos que es prácticamente todo lo que se hace en la segunda parte del proyecto (visualización de datos).

Para ver más sobre la estructura y el funcionamiento de Django ver el [ANEXO I](#).

1.2.3 LIBRERÍA D3.JS

D3.js [2] es una librería de JavaScript que permite manipular documentos con datos. Sobre todo, se usa para realizar visualizaciones de datos. Su nombre se debe a las siglas Data-Driven Documents.

En este proyecto su uso se centra en la visualización de datos mediante gráficos estadísticos. Se ha elegido porque usa las tecnologías estándar HTML, SVG y CSS [3] que nos garantiza que funcionará en cualquier navegador web.

CAPITULO II
REQUISITOS Y
ANÁLISIS
FUNCIONAL

2. REQUISITOS Y ANÁLISIS FUNCIONAL

En este apartado se explicará los requisitos de desarrollo, así como el análisis funcional de las aplicaciones a desarrollar.

2.1 REQUISITOS

Estos requisitos describen cuales deben ser las funcionalidades de los sistemas a desarrollar:

2.1.1 REQUISITOS DE LA APLICACIÓN DE SCRAPEO

1. Realiza escraeos desde tres sitios distintos:
 - a. www.futnavarra.es
 - b. www.gesportsl.com
 - c. Archivo Excel que cumple una plantilla, subido al servidor por el usuario
2. Si se escoge la opción de escraeo de futnavarra hay 4 opciones distintas:
 - a. Escrapear categorías
 - b. Escrapear equipos
 - c. Escrapear tipos de clasificaciones
 - d. Escrapear resultados última jornada y clasificaciones

Si es la primera vez que se realiza el escraeo, se deberá realizar en el orden de los apartados recientemente descritos. Si no es la primera vez, solo habría que escrapear el último apartado, ya que el resto de datos no va a cambiar, a no ser que hayamos cambiado de fase o de temporada, en ese caso conviene reescrapearlo todo.

3. Si escoge la opción de escraeo de gesportsl directamente se escraea todo.
4. Si se escoge la opción de escraeo del Excel nos da la opción de escoger el archivo de nuestro ordenador. Hay que tener en cuenta que debe ser un archivo con extensión de Excel, si no, no nos dará opción de escrapear. Además, debe cumplir la plantilla prefijada, si no, el escraeo será incorrecto. Esta acción sigue el siguiente patrón:
 - a. Aloja el archivo Excel seleccionado en una carpeta del servidor
 - b. Escraea el archivo alojado
 - c. Borra el archivo del servidor
5. Los escraeos no deben duplicar datos. Esto quiere decir que antes de almacenar algo en la base de datos se comprueba que el registro no existe, si existe, lo modifica. Esto solo se incumple en el caso de las categorías, ya que se requiere guardar un histórico para realizar las visualizaciones posteriores.
6. Los escraeos deben ser bloqueantes. Esto quiere decir que mientras estoy escraeando algo no puedo escrapear otra cosa.

2.1.2 REQUISITOS DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS

1. Se visualizarán los datos de los distintos campeonatos:
 - a. Futnavarra
 - b. Intermunicipal
 - c. Fútbol txiki
2. De cada campeonato se quieren visualizar distintos apartados:
 - a. Equipos
 - b. Partidos
 - c. Clasificaciones
 - d. Estadísticas
3. Para tener una visualización más dinámica, se les aplicará una serie de filtros a los apartados. Se podrá filtrar por:
 - a. Temporadas
 - b. Categoría
 - c. Grupo
 - d. Fase
 - e. Fecha (solo en el apartado de Partidos)
4. A su vez los filtros se filtran entre ellos, por ejemplo, en dos fases distintas de la misma categoría no tiene por qué haber los mismos grupos, así que al cambiar la fase cambiarán los grupos.
5. Hay que tener controlada la posibilidad de que no haya datos sobre los equipos, partidos o clasificaciones de una temporada, categoría, grupo y fase concretos.
6. Se podrá buscar por nombre de equipo en el listado de equipos.
7. Se mostrarán estadísticas sobre los distintos equipos:
 - a. Gráfico de barras que muestra el top-10 de categorías de un equipo seleccionado, es decir, las 10 categorías en las que el equipo ha ganado más partidos. Esta solo se mostrará en Futnavarra.
 - b. Calendario de calor que muestra por cada equipo si ha ganado, empatado, perdido o descansado en cada jornada. Esta solo se mostrará en Intermunicipal y en Fútbol Txiki.
 - c. Radar comparando dos equipos. La comparación se realiza mediante las variables de la clasificación.
 - d. Líneas paralelas por cada grupo de equipos que muestra la evolución en la tabla de clasificaciones del equipo en cada jornada.

2.2 ANÁLISIS FUNCIONAL

2.2.1 DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN DE SCRAPEO

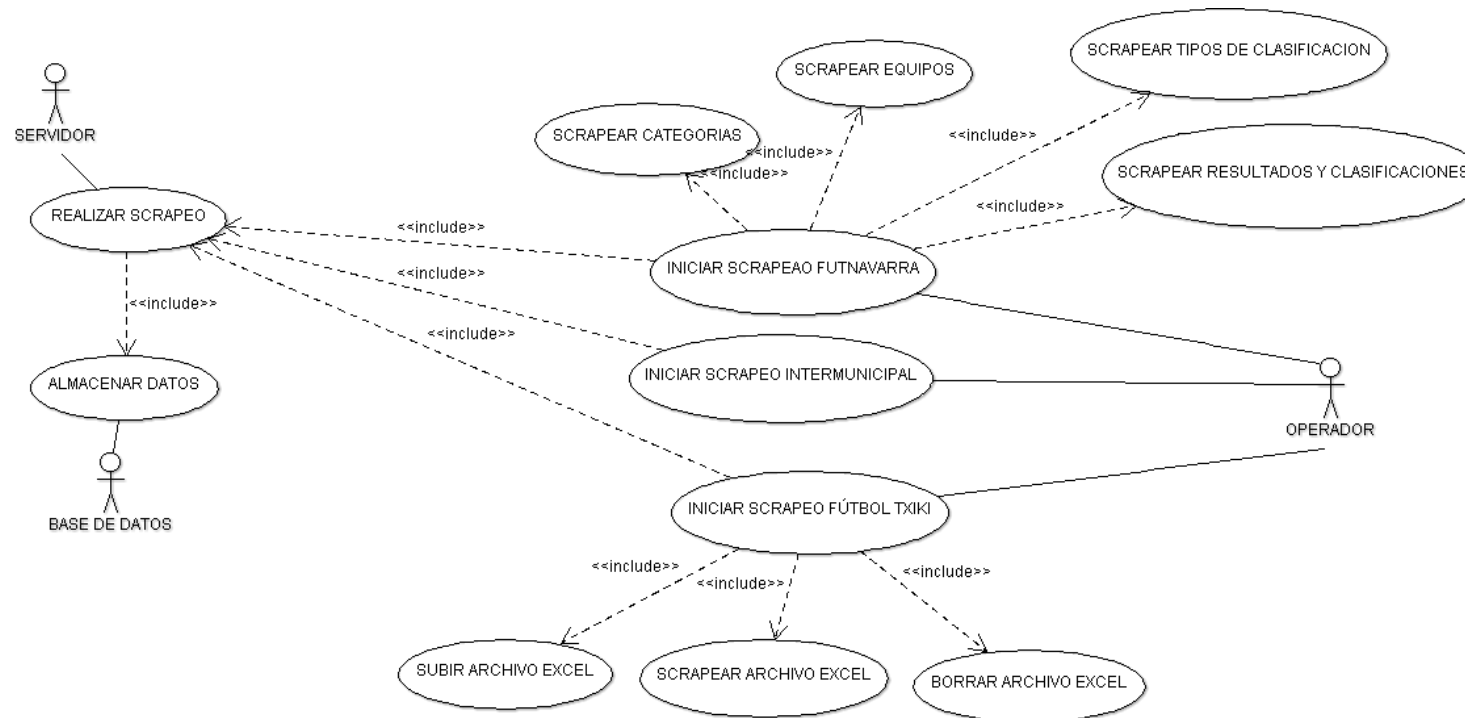


Figura 2.2.1.1 Diagrama de casos de uso de la aplicación de scrapeo

Como se aprecia en el diagrama hay tres actores que interactúan con el sistema:

- Operador: Empleado que se encargará de poner en marcha la aplicación para que comience a realizar los scrapeos.
- Servidor: actúa como servidor. Se encarga de realizar las operaciones de scrapeo demandadas por el operador.
- Base de datos: actúa como base de datos. Donde se almacenarán los datos recogidos del scrapeo.

A continuación, se describirá en detalle cada caso de uso:

- **Caso de uso 1:** Iniciar scrapeo futnavarra (CdU01):
 - Actores: Operario
 - Descripción: El operario ha decidido escrapear la página de futnavarra. Puede elegir que escrapear entre las opciones anteriormente descritas.
 - Precondición: El servidor debe estar funcionando y la página de futnavarra operativa. Recomendado que el día elegido para hacerlo sea el jueves.
 - Flujo básico:
 1. El operador decide que opción de futnavarra va a escrapear.
 2. El servidor recibe la petición y la procesa. El usuario debe esperar mientras se procesa.
 3. El servidor avisa de que la petición ha sido satisfecha.
 - Flujo alternativo:

Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
 - Postcondición:

Los datos escrapeados están almacenados en la base de datos.
 - Condiciones de error:

Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
 - Criterio de aceptación:

Los datos de las urls escrapeadas correctamente están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 2:** Iniciar scrapeo intermunicipal (CdU02):

- Actores: Operario
- Descripción: El operario ha decidido escrapear la página del campeonato intermunicipal.
- Precondición: El servidor debe estar funcionando y la página de gesportsl operativa. Recomendado que el día elegido para hacerlo sea el jueves.
- Flujo básico:
 1. El operador ha decidido escrapear el torneo intermunicipal.
 2. El servidor recibe la petición y la procesa. El usuario debe esperar mientras se procesa.
 3. El servidor avisa de que la petición ha sido satisfecha.
- Flujo alternativo:

Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Postcondición:

Los datos escrapeados están almacenados en la base de datos.
- Condiciones de error:

Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Criterio de aceptación:

Los datos de las urls escrapeadas correctamente están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 3:** Iniciar scrapeo fútbol txiki (CdU03):

- Actores: Operario
- Descripción: El operario ha decidido escrapear el campeonato fútbol txiki.
- Precondición: El servidor debe estar funcionando y el archivo Excel cumplir la plantilla prefijada
- Flujo básico:
 1. El operador ha decidido escrapear el torneo de fútbol txiki.
 2. El usuario elige un archivo del ordenador para subir al servidor.
 3. El archivo es subido al servidor.
 4. El servidor recibe la petición y la procesa. El servidor lee el archivo Excel y almacena los datos leídos, tras esto lo borra. El usuario debe esperar mientras se procesa.
 5. El servidor avisa de que la petición ha sido satisfecha.
- Flujo alternativo:

Si el archivo subido no es un Excel se avisa y no se deja escrapear.

- Postcondición:
Los datos escrapeados están almacenados en la base de datos.
- Condiciones de error:
Si el archivo subido no es un Excel se avisa y no se deja escrapear.
- Criterio de aceptación:
Los datos leídos están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 4:** Scrapear categorías (CdU04):

- Actores: Operario
- Descripción: El operario ha decidido escrapear futnavarra, y dentro de las opciones ha decidido escrapear las categorías.
- Precondición: El servidor debe estar funcionando y el operador ha elegido la opción de categorías.
- Flujo básico:
 1. El operador ha decidido escrapear futnavarra y dentro de esta la opción de categorías.
 2. El servidor recibe la petición y la procesa. El usuario debe esperar mientras se procesa.
 3. El servidor avisa de que la petición ha sido satisfecha.
- Flujo alternativo:
Si alguna de las urls a escrapear falla hay una aviso y opción de resrapear.
- Postcondición:
Los datos escrapeados están almacenados en la base de datos.
- Condiciones de error:
Si alguna de las urls a escrapear falla hay una aviso y opción de resrapear.
- Criterio de aceptación:
Los datos escrapeados están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 5:** Scrapear equipos (CdU05):

- Actores: Operario
- Descripción: El operario ha decidido escrapear futnavarra, y dentro de las opciones ha decidido escrapear los equipos.
- Precondición: El servidor debe estar funcionando y el operador ha elegido la opción de equipos.

- Flujo básico:
 1. El operador ha decidido escrapear futnavarra y dentro de esta la opción de equipos.
 2. El servidor recibe la petición y la procesa. El usuario debe esperar mientras se procesa.
 3. El servidor avisa de que la petición ha sido satisfecha.
- Flujo alternativo:
Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Postcondición:
Los datos escrapeados están almacenados en la base de datos.
- Condiciones de error:
Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Criterio de aceptación:
Los datos escrapeados están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 6:** Scrapear tipos de clasificación (CdU06):

- Actores: Operario
- Descripción: El operario ha decidido escrapear futnavarra, y dentro de las opciones ha decidido escrapear los tipos de clasificación.
- Precondición: El servidor debe estar funcionando y el operador ha elegido la opción de tipos de clasificación.
- Flujo básico:
 1. El operador ha decidido escrapear futnavarra y dentro de esta la opción de tipos de clasificación.
 2. El servidor recibe la petición y la procesa. El usuario debe esperar mientras se procesa.
 3. El servidor avisa de que la petición ha sido satisfecha.
- Flujo alternativo:
Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Postcondición:
Los datos escrapeados están almacenados en la base de datos.
- Condiciones de error:
Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Criterio de aceptación:
Los datos escrapeados están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 7:** Scrapear equipos (CdU07):

- Actores: Operario
- Descripción: El operario ha decidido escrapear futnavarra, y dentro de las opciones ha decidido escrapear los resultados y las clasificaciones.
- Precondición: El servidor debe estar funcionando y el operador ha elegido la opción de resultados y clasificación.
- Flujo básico:
 1. El operador ha decidido escrapear futnavarra y dentro de esta la opción de resultados y clasificación.
 2. El servidor recibe la petición y la procesa. El usuario debe esperar mientras se procesa.
 3. El servidor avisa de que la petición ha sido satisfecha.
- Flujo alternativo:

Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Postcondición:

Los datos escrapeados están almacenados en la base de datos.
- Condiciones de error:

Si alguna de las urls a escrapear falla hay una aviso y opción de rescrapear.
- Criterio de aceptación:

Los datos escrapeados están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 8:** Realizar escraqueo (CdU08):

- Actores: Servidor
- Descripción: El operario ha decidido escrapear una de las opciones, le ha mandado una petición al servidor y este debe satisfacerla.
- Precondición: El servidor debe estar funcionando y el operador ha decidido iniciar uno de los escraeos.
- Flujo básico:
 1. El operador ha decidido escrapear una de las opciones y manda una petición al servidor.
 2. El servidor recibe la petición y la procesa.
 3. Una vez realizada la petición avisa de que la petición ha sido satisfecha.
- Flujo alternativo:

Si la petición no puede ser satisfecha se manda un mensaje de error.

- Postcondición:
El servidor ha satisfecho la petición recibida y ha almacenado los datos en la base de datos.
- Condiciones de error:
Si la petición no puede ser satisfecha se manda un mensaje de error.
- Criterio de aceptación:
Los datos escrapeados están almacenados de una forma adecuada en la base de datos.

➤ **Caso de uso 9:** Almacenar datos (CdU09):

- Actores: Base de datos
- Descripción: Durante la realización de las peticiones el servidor se conectará a la base de datos correspondiente y hará una serie de consultas a esta. La mayoría de las consultas serán de tipo INSERT.
- Precondición: El servidor debe estar realizando una petición.
- Flujo básico:
 1. El servidor está realizando una petición y se ha conectado a una de las bases de datos.
 2. El servidor realiza una consulta SELECT para ver si los datos a almacenar existen. Si no existe un registro con estos datos:
 - a. El servidor realiza una consulta INSERT.
 Si existe:
 - b. El servidor realiza una consulta UPDATE.
 3. Las consultas se han realizado con éxito.
- Flujo alternativo:
Si ha habido un error en alguna de las consultas el dato no habrá sido almacenado o modificado.
- Postcondición:
Los datos han sido almacenados o modificados correctamente.
- Condiciones de error:
Si ha habido un error en alguna de las consultas el dato no habrá sido almacenado o modificado.
- Criterio de aceptación:
Los datos han sido almacenados o modificados correctamente.

2.2.2 DIAGRAMA DE ACTIVIDAD DE LA APLICACIÓN DE SCRAPEO

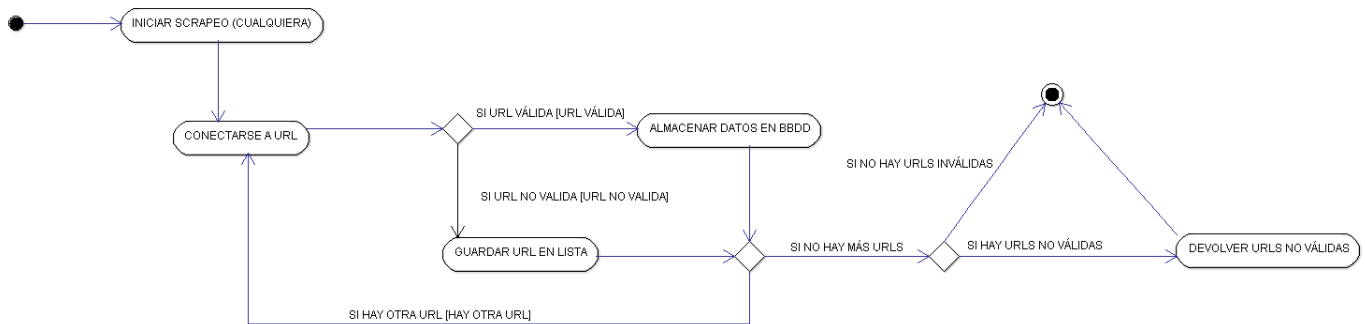


Figura 2.2.2.1 Diagrama de actividad de uso de la aplicación de escraqueo

Cuando el usuario quiere realizar cualquiera de los escraeos internamente el servidor realiza la siguiente secuencia de acciones:

1. Primero se conecta a la url que corresponda. En el caso de futnavarra dependiendo de la opción que se haya escogido (categorías, equipos, tipos de clasificación resultados y clasificaciones) y la temporada en la que estemos se conectará a un conjunto de urls determinado. En el caso de intermunicipal el conjunto de urls dependerá de la temporada en la que estemos.
2. El servidor comprueba si la url es válida haciendo una petición y comprobando el atributo del estado. Si el estado es 200 la url es válida si no, no lo es:
 - a. Una vez comprobada que la url es válida se procede a leer de la web los datos a almacenar, se les da formato y se meten en la base de datos.
 - b. Si la url no es válida se mete en una lista de urls no válidas.
3. Si hay más urls a scrapear se vuelve al paso 2.
4. Si hay urls en la lista de urls no válidas se muestra un mensaje con estas urls y se acaba, si no las hay se acaba.

Esta secuencia se muestra también en el siguiente diagrama de secuencia:

2.2.3 DIAGRAMA DE SECUENCIA DE LA APLICACIÓN DE SCRAPEO

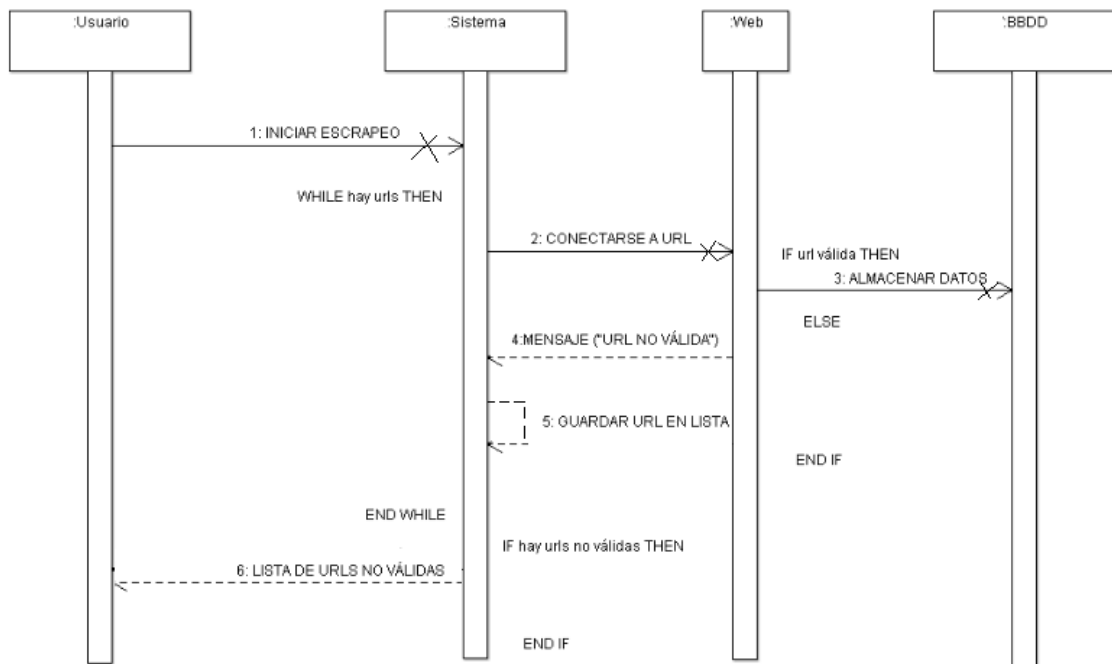


Figura 2.2.3.1 Diagrama de secuencia de la aplicación de escraqueo

2.2.4 DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS

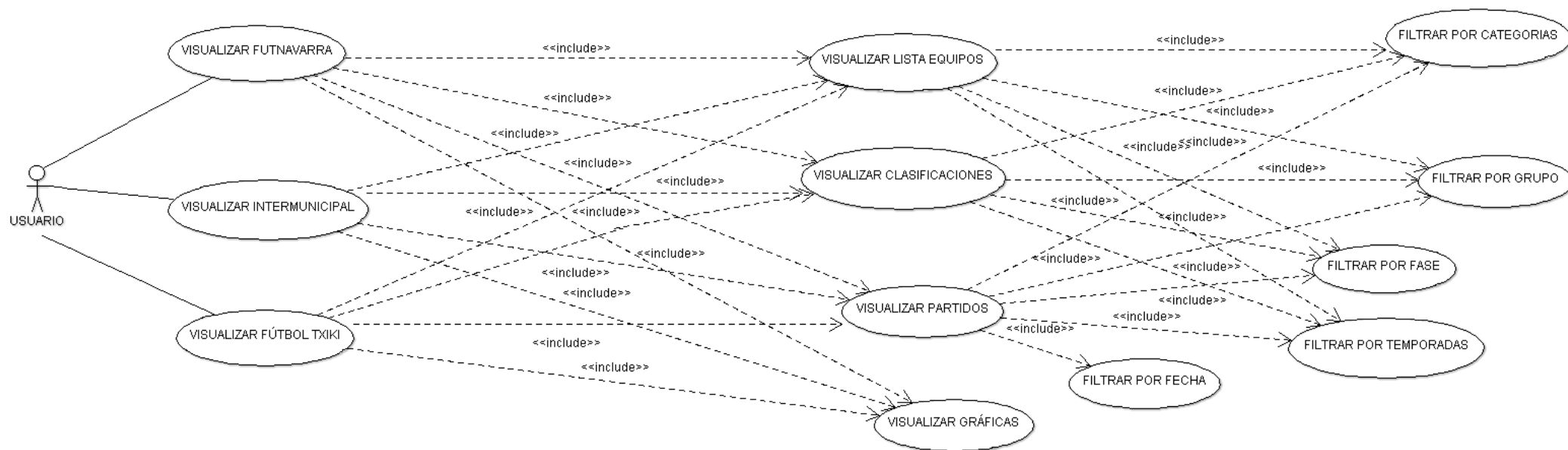


Figura 2.2.4.1 Diagrama de casos de uso de la aplicación de visualización de datos

Como se aprecia en el diagrama hay un solo actor que interactúa con el sistema:

- **Usuario:** Usuario externo que va a ver los datos de la página.
- **Servidor:** actúa como servidor. Se encarga de realizar las operaciones de visualización demandadas por el usuario.

A continuación, se describirá en detalle cada caso de uso:

- **Caso de uso 1:** Visualizar futnabarra (CdU01):
 - Actores: Usuario, servidor
 - Descripción: El usuario ha decidido visualizar la página de futnabarra. Tiene opción de visualizar los equipos, los partidos, las clasificaciones o las gráficas.
 - Precondición: El servidor debe estar funcionando.
 - Flujo básico:
 1. El usuario decide que opción de futnabarra va a visualizar.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
 - Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
 - Postcondición:

El usuario visualiza lo que ha solicitado.
 - Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
 - Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.
- **Caso de uso 2:** Visualizar intermunicipal (CdU02):
 - Actores: Usuario, servidor
 - Descripción: El usuario ha decidido visualizar la página del campeonato intermunicipal. Tiene opción de visualizar los equipos, los partidos, las clasificaciones o las gráficas.
 - Precondición: El servidor debe estar funcionando.
 - Flujo básico:
 1. El usuario decide que opción del campeonato intermunicipal va a visualizar.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.

- Flujo alternativo:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:
El usuario visualiza lo que ha solicitado.
- Condiciones de error:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:
El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 3:** Visualizar fútbol txiki (CdU03):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido visualizar la página de fútbol txiki. Tiene opción de visualizar los equipos, los partidos, las clasificaciones o las gráficas.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que opción de fútbol txiki va a visualizar.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:
El usuario visualiza lo que ha solicitado.
- Condiciones de error:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:
El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 4:** Visualizar lista equipos (CdU04):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido visualizar la lista de equipos de uno de los campeonatos.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a visualizar la lista de los equipos de un campeonato.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.

- Flujo alternativo:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:
El usuario visualiza lo que ha solicitado.
- Condiciones de error:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:
El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 5:** Visualizar clasificaciones (CdU05):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido visualizar las clasificaciones de uno de los campeonatos.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a visualizar las clasificaciones de un campeonato.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:
El usuario visualiza lo que ha solicitado.
- Condiciones de error:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:
El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 6:** Visualizar partidos (CdU06):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido visualizar los partidos de uno de los campeonatos.
- Precondición: El servidor debe estar funcionando.

- Flujo básico:
 1. El usuario decide que va a visualizar los partidos de un campeonato.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:

El usuario visualiza lo que ha solicitado.
- Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 7: Visualizar gráficas (CdU07):**

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido visualizar las gráficas de uno de los campeonatos.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a visualizar las gráficas de un campeonato.
 2. El usuario decide que gráfica va a visualizar.
 3. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:

El usuario visualiza lo que ha solicitado.
- Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 8:** Filtrar por categoría (CdU08):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido filtrar la búsqueda por categorías y el servidor deberá devolverle los datos pertinentes.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a filtrar los datos por categorías.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:

El usuario visualiza lo que ha solicitado.
- Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 9:** Filtrar por grupo (CdU09):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido filtrar la búsqueda por grupo y el servidor deberá devolverle los datos pertinentes.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a filtrar los datos por grupo.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:

El usuario visualiza lo que ha solicitado.
- Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 10:** Filtrar por fase (CdU10):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido filtrar la búsqueda por fase y el servidor deberá devolverle los datos pertinentes.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a filtrar los datos por fase.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:

El usuario visualiza lo que ha solicitado.
- Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 11:** Filtrar por temporada (CdU11):

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido filtrar la búsqueda por temporada y el servidor deberá devolverle los datos pertinentes.
- Precondición: El servidor debe estar funcionando.
- Flujo básico:
 1. El usuario decide que va a filtrar los datos por temporada.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:

El usuario visualiza lo que ha solicitado.
- Condiciones de error:

Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:

El servidor ha contestado a la petición de visualización con los datos correctos.

➤ **Caso de uso 12: Filtrar por fecha (CdU12):**

- Actores: Usuario, servidor
- Descripción: El usuario ha decidido filtrar la búsqueda por fecha y el servidor deberá devolverle los datos pertinentes.
- Precondición: El servidor debe estar funcionando. El usuario está en la pantalla de partidos.
- Flujo básico:
 1. El usuario decide que va a filtrar los datos por temporada.
 2. El servidor recibe la petición y muestra el usuario los datos que ha solicitado.
- Flujo alternativo:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Postcondición:
El usuario visualiza lo que ha solicitado.
- Condiciones de error:
Si no hay datos disponibles se mostrará una pantalla de no hay datos.
- Criterio de aceptación:
El servidor ha contestado a la petición de visualización con los datos correctos.

2.2.5 DIAGRAMA DE ACTIVIDAD DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS

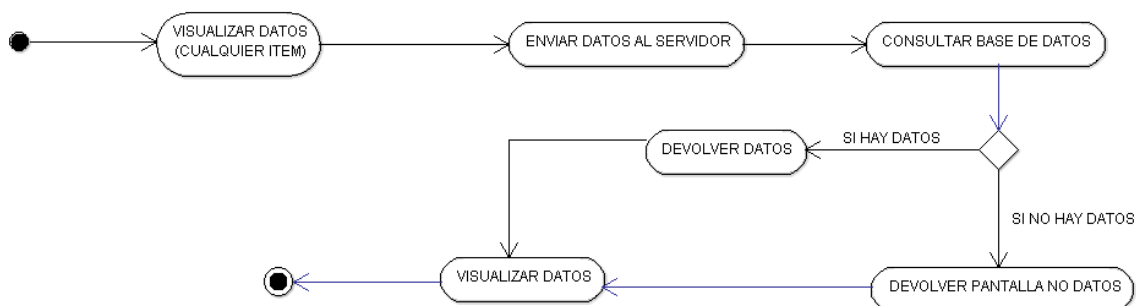


Figura 2.2.5.1 Diagrama de actividad de la aplicación de visualización de datos

Cuando el usuario quiere visualizar cualquier dato internamente el servidor realiza la siguiente secuencia de acciones:

1. El usuario decide que datos quiere visualizar y se manda una petición al servidor mediante una url.
2. El servidor recibe la petición y busca en la base de datos correspondiente los datos a visualizar.
3. Si en la base de datos existen datos se devolverán estos datos y el usuario los visualizará.
4. Si en la base de datos no existen datos se devolverá al usuario una pantalla informándole de que no hay datos disponibles.

Esta secuencia se muestra también en el siguiente diagrama de secuencia:

2.2.6 DIAGRAMA DE SECUENCIA DE LA APLICACIÓN DE VISUALIZACIÓN DE DATOS

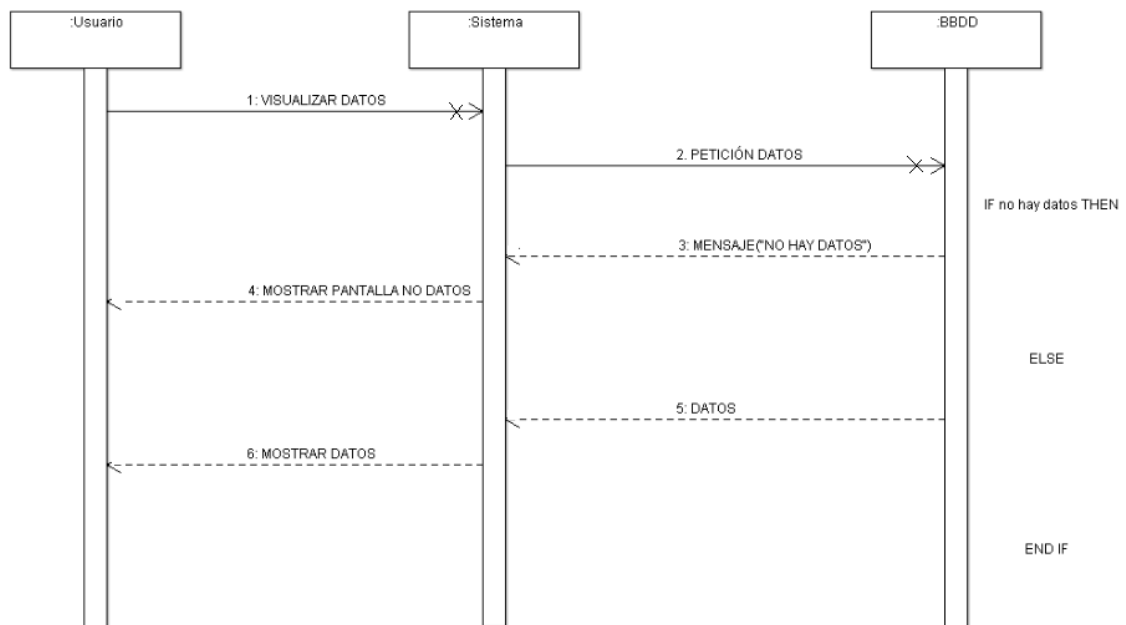


Figura 2.2.6.1 Diagrama de secuencia de la aplicación de visualización de datos

2.3 ANÁLISIS BASES DE DATOS

Para la realización de las dos aplicaciones son necesarias tres bases datos (una por cada campeonato), para almacenar y consultar los datos:

1. Futnavarra → que almacenará los datos necesarios para visualizar los datos de la federación navarra de fútbol.
2. Intermunicipal → que almacenará los datos necesarios para visualizar los datos del tornea intermunicipal de fútbol.
3. Fútbol Txiki → que almacenará los datos necesarios para visualizar la información del torneo de fútbol txiki.

En primer lugar, se explica el análisis para la base de datos de Futnavarra:

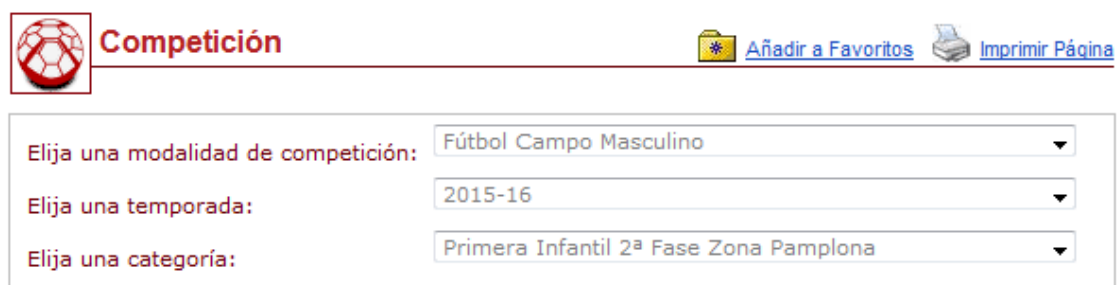
2.3.1 BBDD FUTNAVARRA

1. DESCRIPCIÓN:

Futnavarra engloba todos los datos de las competiciones de fútbol base de Navarra. Entre estas competiciones se puede diferenciar entre fútbol en campo y fútbol sala. Además, dentro de estas podemos diferenciar entre dos sexos, masculino y femenino. Queda claro que nos quedan cuatro combinaciones, a partir de ahora las llamaremos juegos, que son:

1. Fútbol campo masculino
2. Fútbol campo femenino
3. Fútbol sala masculino
4. Fútbol sala femenino

Cada uno de estos juegos tiene una serie de categorías, en este caso solo nos interesan las correspondientes a deporte base. Un equipo solo puede pertenecer a una categoría y una categoría puede ser de varios equipos. Dentro de las categorías existen fases, como mínimo una, como máximo dos. Hay que tener en cuenta que en algunos casos la categoría en sí ya es una fase, por ejemplo, la categoría “Primera Infantil 2ª fase Zona Pamplona” por su nombre nos indica que ya es de la fase 2. Otras categorías tendrán dos fases, por ejemplo, la categoría “Segunda Infantil – Juegos Deportivos de Navarra” tendrá fase 1 y fase 2.




Competición



Elija una modalidad de competición: Fútbol Campo Masculino

Elija una temporada: 2015-16

Elija una categoría: Primera Infantil 2ª Fase Zona Pamplona

Figura 2.3.1.1 Ejemplo de categoría que ya es fase 2


Competición


[Añadir a Favoritos](#)

[Imprimir Página](#)

Elija una modalidad de competición:
Fútbol Campo Masculino

Elija una temporada:
2015-16

Elija una categoría:
Segunda Infantil - Juegos Deportivos de Navarra

Elija una fase:
1ª fase
-- Selecciona fase --
1ª fase
2ª fase

Figura 2.3.1.2 Ejemplo de categoría que tiene 2 fases

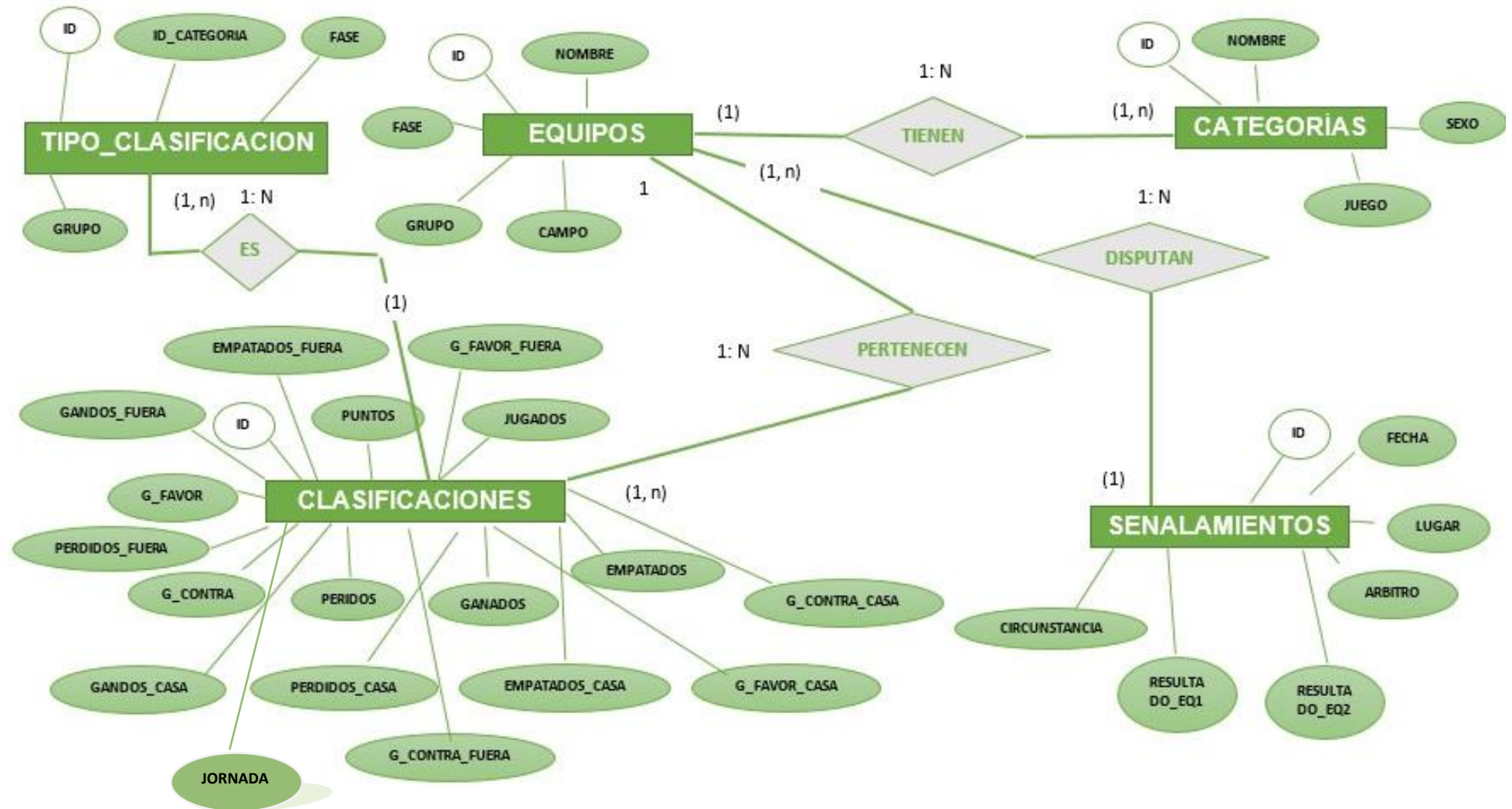
De cada categoría se necesita recoger el id (será el que se usa en la página futnavarra), el nombre, el juego al que pertenecen (campo 'C', sala 'S'), y el sexo (masculino 'M', femenino 'F'). Como los equipos pertenecientes a una categoría, puede que estén en una fase, pero no en otra, se ha decidido almacenar el atributo 'fase' en la entidad 'EQUIPOS'.

De cada equipo se requiere recoger información sobre su nombre, (cuál es su campo de juego), a que categoría pertenecen, a que fase y a que temporada. Además, dentro de cada categoría el equipo pertenecerá a un grupo en concreto que también se deberá almacenar.

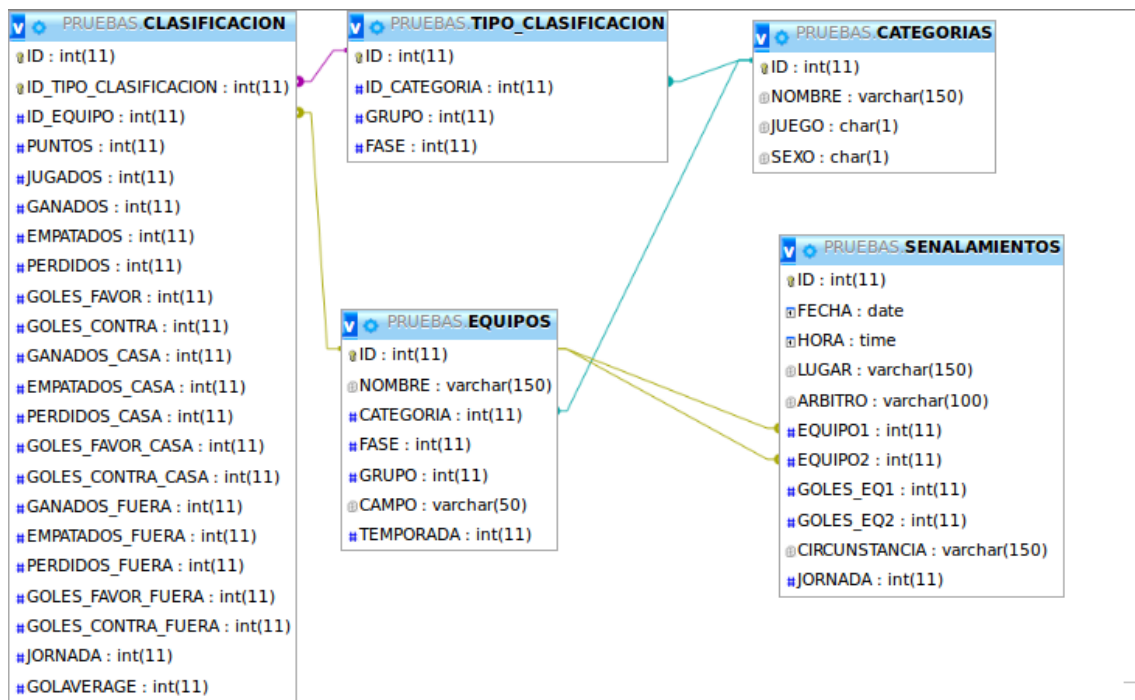
Dentro de la competición existen dos entidades, 'SEÑALAMIENTOS' y 'PARTIDOS'. De los señalamientos se requiere guardar la fecha, el lugar, el árbitro, y que dos equipos van a disputar el partido. De los partidos, la fecha, el lugar, que equipos van a disputarlo y el resultado. Como esta información se recoge de un solo escraqueo se ha decidido solo almacenar una entidad, señalamientos, ya que cuando el señalamiento se ha disputado, nos da el resultado de este, así que en esta entidad tendremos todos los datos necesarios de ambas. Entonces, tendremos una entidad llamada 'SEÑALAMIENTOS' con los atributos fecha, lugar, árbitro, equipo1, equipo2, y resultado. Esta entidad se relaciona con la de 'EQUIPOS' de forma que 1 equipo está en varios señalamientos y en un señalamiento hay dos equipos. Es necesario un atributo adicional llamado circunstancia que nos informe sobre si el partido ha sido aplazado, suspendido, invertido o anulado.

Por último, los equipos de cada categoría, fase y grupo pertenecen a una clasificación. En la clasificación se desea almacenar el id del equipo, cuantos puntos tiene, cuantos partidos a jugado, cuantos ganados, perdidos y empatados, así como los goles a favor y en contra que tiene. También cuantos partidos ha ganado, perdido y empatado con goles a favor y en contra, en casa y fuera. Es necesario saber a qué jornada hace referencia la clasificación para poder guardar un histórico. Un equipo pertenece a una clasificación y a una clasificación pertenecen varios equipos. Como se ha mencionado antes los equipos de cada categoría, fase y grupo pertenecen a una clasificación, hay que crear una entidad 'TIPO_CLASIFICACION' que almacene estos datos.

2. MODELO ENTIDAD - RELACIÓN:



3. MODELO PASADO A TABLAS:



4. ANÁLISIS DE LAS TABLAS¹:

➤ CATEGORÍAS:

De cada categoría se necesita recoger el id (será el que se usa en la página futnavarra), el nombre, el juego al que pertenecen (campo 'C', sala 'S'), el sexo (masculino 'M', femenino 'F'), y la edad. Entonces, la tabla categorías tendrá los siguientes atributos:

CATEGORÍAS (ID, NOMBRE, JUEGO, SEXO, EDAD)

➤ EQUIPOS:

De cada equipo se requiere recoger información sobre su nombre, cuál es su campo de juego, a que categoría pertenecen, a que fase y a que temporada. Además, dentro de cada categoría el equipo pertenecerá a un grupo en concreto que también se deberá almacenar.

EQUIPOS (ID, NOMBRE, CAMPO, ID_CATEGORIA, FASE, GRUPO, TEMPORADA)

⇒ ID_CATEGORIA hace referencia al atributo ID de la entidad CATEGORIAS.

¹ Los atributos subrayados con línea continua son las claves primarias. Los subrayados con línea discontinua son las claves foráneas.

➤ **SEÑALAMIENTOS:**

Tendremos una entidad llamada 'SENALAMIENTOS' con los atributos fecha, lugar, árbitro, equipo1, equipo2, y resultado. Es necesario un atributo adicional llamado circunstancia que nos informe sobre si el partido ha sido aplazado, suspendido, invertido o anulado.

SENALAMIENTOS (ID, FECHA, LUGAR, ARBITRO, EQUIPO1, EQUIPO2, RESULTADO_EQ1, RESULTADO_EQ2, CIRCUNSTANCIA)

⇒ EQUIPO1 y EQUIPO2 hacen referencia al atributo ID de la entidad EQUIPOS.

➤ **TIPO_CLASIFICACION:**

Los equipos de cada categoría, fase y grupo pertenecen a una clasificación, hay que crear una entidad 'TIPO_CLASIFICACION' que almacene estos datos.

TIPO_CLASIFICACION (ID, ID_CATEGORIA, GRUPO, FASE)

⇒ ID_CATEGORIA hace referencia al atributo ID de la entidad CATEGORIAS.

➤ **CLASIFICACIONES:**

En la clasificación se desea almacenar el nombre del equipo, cuantos puntos tiene, cuantos partidos a jugado, cuantos ganados, perdido y empatado, así como los goles a favor y en contra que tiene. También cuantos partidos ha ganado, perdido y empatado con goles a favor y en contra, en casa y fuera. Es necesario saber a qué jornada hace referencia la clasificación para poder guardar un histórico.

CLASIFICACIONES (ID, ID_TIPO_CLASIFICACIÓN, ID_EQUIPO, PUNTOS, JUGADOS, GANADOS, EMPATADOS, PERDIDOS, G_FAVOR, G_CONTRA, GANADOS_CASA, EMPATADOS_CASA, PERDIDOS_CASA, G_FAVOR_CASA, G_CONTRA_CASA, GANADOS_FUERA, EMPATADOS_FUERA, PERDIDOS_FUERA, G_FAVOR_FUERA, G_CONTRA_FUERA, JORNADA)

⇒ ID_TIPO_CLASIFICACION hace referencia al atributo ID de la entidad TIPO_CLASIFICACION.

⇒ ID_EQUIPO hace referencia al atributo ID de la entidad EQUIPOS.

2.3.2 BBDD INTERMUNICIPAL

1. DESCRIPCIÓN:

En esta BBDD se almacenarán todos los datos necesarios sobre el torneo intermunicipal de Navarra. Este torneo tiene 1º y 2º vuelta. Como todo torneo hay equipos que lo disputan que se dividen en categorías y dentro de estas en grupos.

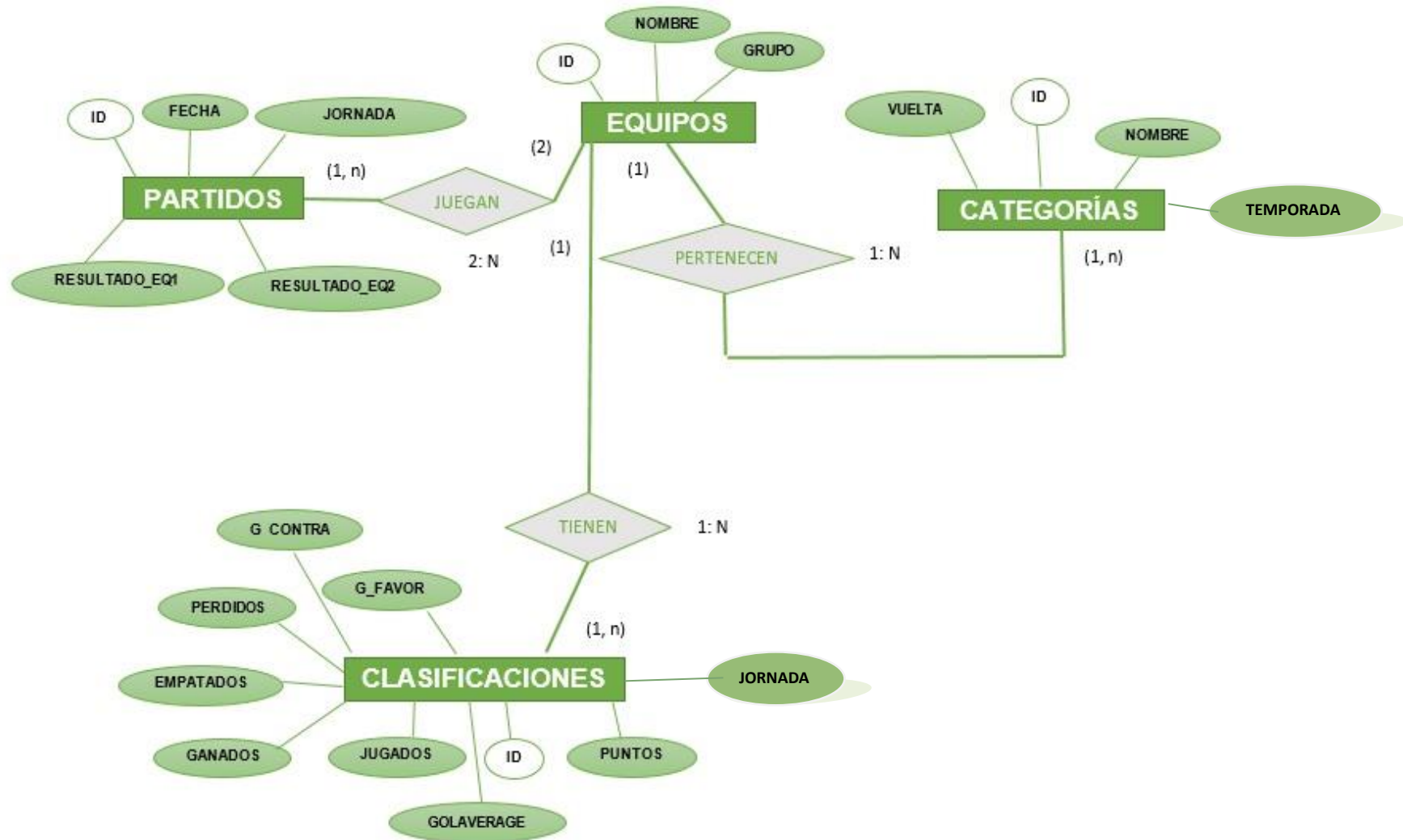
Necesitamos guardar información de las categorías, en concreto un id para identificarlas y su nombre. Como el torneo tienes dos vueltas, las categorías estarán en las dos así que se ha decidido almacenar este campo también. Está claro que cada categoría pertenece también a una temporada concreta.

También tendremos que guardar los datos necesarios sobre los equipos, que son su nombre, y el grupo al que pertenecen dentro de una categoría. Un equipo pertenece a una sola categoría y a una categoría pertenecen varios equipos.

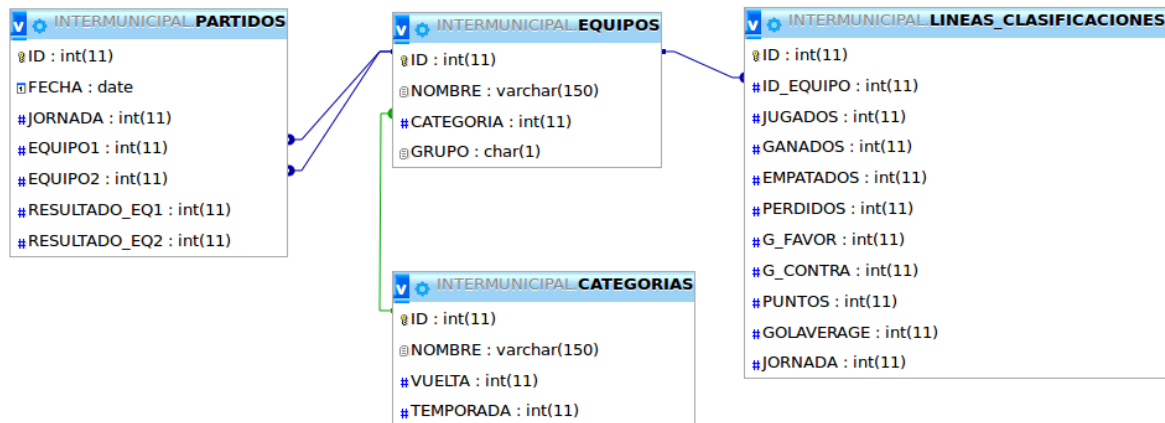
Los equipos disputarán partidos de los cuales queremos la siguiente información: el id del partido para identificarlo, la jornada, la fecha, que equipos lo disputan, y el resultado. Un partido será disputado por dos equipos y un equipo puede disputar varios partidos.

Como es un torneo habrá una clasificación por cada categoría y grupo. Tendremos una entidad 'LINEAS_CLASIFICACIONES' con los atributos id, id del equipo, cuantos partidos a jugado, ganado, empatado y perdido, cuantos goles a favor y en contra tiene, su número de puntos y el golaverage. Como se desea guardar un histórico de la clasificación, es necesario almacenar a que jornada pertenece.

2. MODELO ENTIDAD - RELACIÓN:



3. MODELO PASADO A TABLAS:



4. ANÁLISIS DE LAS TABLAS²:

➤ CATEGORÍAS:

Necesitamos guardar información de las categorías, en concreto un id para identificarlas y su nombre. Como el torneo tienes dos vueltas, las categorías estarán en las dos así que se ha decidido almacenar este campo también. Está claro que cada categoría pertenece también a una temporada concreta.

CATEGORÍAS (ID, NOMBRE, VUELTA, TEMPORADA)

➤ EQUIPOS:

También tendremos que guardar los datos necesarios sobre los equipos, que son su nombre, y el grupo al que pertenecen dentro de una categoría. Un equipo pertenece a una sola categoría y a una categoría pertenecen varios equipos.

EQUIPOS (ID, NOMBRE, ID_CATEGORIA, GRUPO)

⇒ ID_CATEGORIA hace referencia al atributo ID de la entidad CATEGORIAS.

² Los atributos subrayados con línea continua son las claves primarias. Los subrayados con línea discontinua son las claves foráneas.

➤ **PARTIDOS:**

Los equipos disputarán partidos de los cuales queremos la siguiente información: el id del partido para identificarlo, la fecha, la jornada, que equipos lo disputan, y el resultado.

PARTIDOS (ID, FECHA, JORNADA, EQUIPO1, EQUIPO2, RESULTADO_EQ1, RESULTADO_EQ2)

⇒ EQUIPO1 y EQUIPO2 hacen referencia al atributo ID de la entidad EQUIPOS.

➤ **CLASIFICACIONES:**

Tendremos una entidad 'LINEAS_CLASIFICACIONES' con los atributos id, id del equipo, cuantos partidos a jugado, ganado, empatado y perdido, cuantos goles a favor y en contra tiene, su número de puntos y el golaverage. Como se desea guardar un histórico de la clasificación, es necesario almacenar a que jornada pertenece.

CLASIFICACIONES (ID, ID_EQUIPO, JUGADOS, GANADOS, EMPATADOS, PERDIDOS, GOLES_FAVOR, GOLES_CONTRA, PUNTOS, GOLAVERAGE)

⇒ ID_EQUIPO hace referencia al atributo ID de la entidad EQUIPOS.

2.3.3 BBDD FÚTBOL TXIKI

1. DESCRIPCIÓN:

En esta BBDD se almacenarán todos los datos necesarios sobre el torneo fútbol txiki de Navarra. Este torneo tiene 1º y 2º fase. Como todo torneo hay equipos que lo disputan que se dividen en categorías y dentro de estas en grupos.

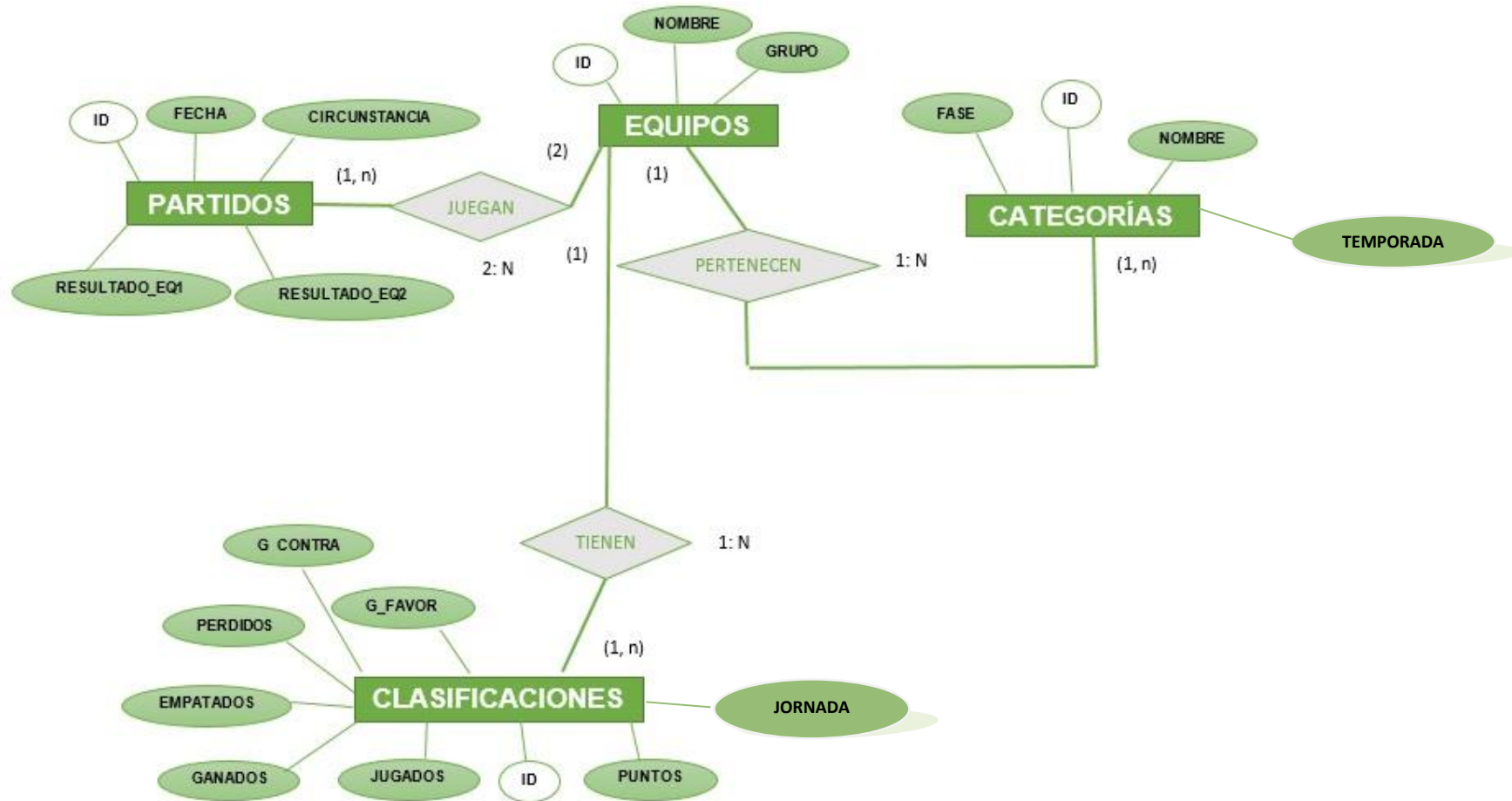
Necesitamos guardar información de las categorías, en concreto un id para identificarlas y su nombre. Como el torneo tienes dos fases, las categorías estarán en las dos así que se ha decidido almacenar este campo también. Está claro que cada categoría pertenece también a una temporada concreta.

También tendremos que guardar los datos necesarios sobre los equipos, que son su nombre, y el grupo al que pertenecen dentro de una categoría. Un equipo pertenece a una sola categoría y a una categoría pertenecen varios equipos.

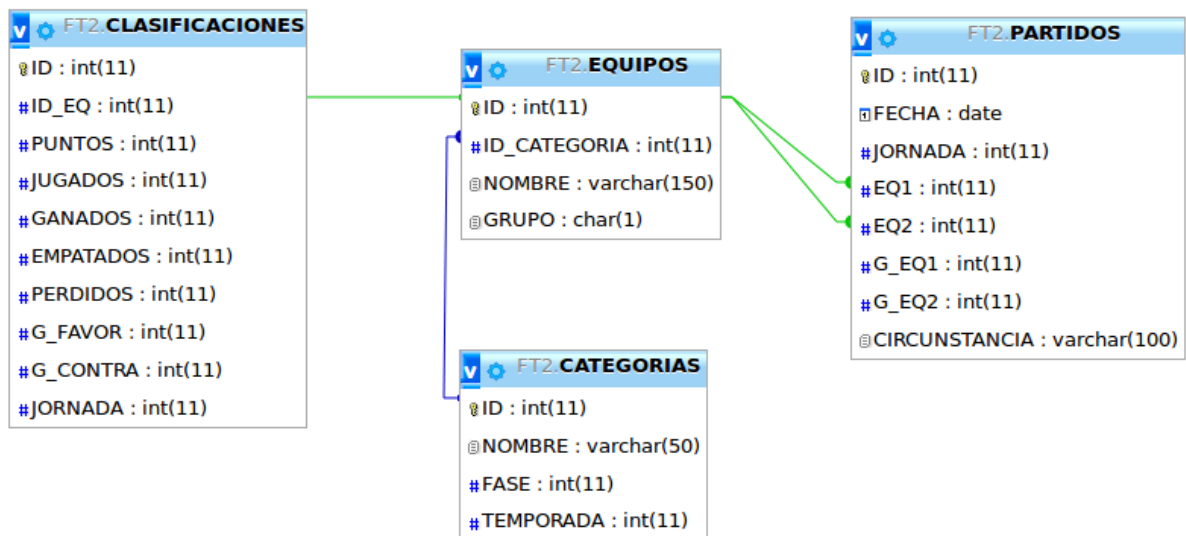
Los equipos disputarán partidos de los cuales queremos la siguiente información: el id del partido para identificarlo, la fecha, que equipos lo disputan, y el resultado. También pueden darse circunstancias especiales en un partido como, por ejemplo, que haya sido aplazado, esto se almacenará en un campo llamado circunstancia. Un partido será disputado por dos equipos y un equipo puede disputar varios partidos.

Como es un torneo habrá una clasificación por cada categoría y grupo. Tendremos una entidad 'CLASIFICACIONES' con los atributos id, id del equipo, cuantos partidos a jugado, ganado, empatado y perdido, cuantos goles a favor y en contra tiene, y su número de puntos. Como se desea guardar un histórico de la clasificación, es necesario almacenar a que jornada pertenece.

2. MODELO ENTIDAD - RELACIÓN:



3. MODELO PASADO A TABLAS:



4. ANÁLISIS DE LAS TABLAS³:

➤ CATEGORÍAS:

Necesitamos guardar información de las categorías, en concreto un id para identificarlas y su nombre. Como el torneo tienes dos vueltas, las categorías estarán en las dos así que se ha decidido almacenar este campo también. Está claro que cada categoría pertenece también a una temporada concreta.

CATEGORÍAS (ID, NOMBRE, FASE, TEMPORADA)

➤ EQUIPOS:

También tendremos que guardar los datos necesarios sobre los equipos, que son su nombre, y el grupo al que pertenecen dentro de una categoría. Un equipo pertenece a una sola categoría y a una categoría pertenecen varios equipos.

EQUIPOS (ID, NOMBRE, ID_CATEGORIA, GRUPO)

⇒ ID_CATEGORIA hace referencia al atributo ID de la entidad CATEGORIAS.

³ Los atributos subrayados con línea continua son las claves primarias. Los subrayados con línea discontinua son las claves foráneas.

➤ **PARTIDOS:**

Los equipos disputarán partidos de los cuales queremos la siguiente información: el id del partido para identificarlo, la fecha, que equipos lo disputan, y el resultado. También pueden darse circunstancias especiales en un partido como, por ejemplo, que haya sido aplazado, esto se almacenará en un campo llamado circunstancia. Un partido será disputado por dos equipos y un equipo puede disputar varios partidos.

PARTIDOS (ID, FECHA, EQUIPO1, EQUIPO2, RESULTADO_EQ1,
RESULTADO_EQ2, CIRCUNSTANCIA)

⇒ EQUIPO1 y EQUIPO2 hacen referencia al atributo ID de la entidad EQUIPOS.

➤ **CLASIFICACIONES:**

Tendremos una entidad 'CLASIFICACIONES' con los atributos id, id del equipo, cuantos partidos a jugado, ganado, empatado y perdido, cuantos goles a favor y en contra tiene, y su número de puntos. Como se desea guardar un histórico de la clasificación, es necesario almacenar a que jornada pertenece.

CLASIFICACIONES (ID, ID_EQUIPO, JUGADOS, GANADOS, EMPATADOS,
PERDIDOS, GOLES_FAVOR, GOLES_CONTRA, PUNTOS, JORNADA)

⇒ ID_EQUIPO hace referencia al atributo ID de la entidad EQUIPOS.

CAPITULO III

MANUAL DE USUARIO

3. MANUAL DE USUARIO

En esta sección se procederá a la presentación del manual del usuario en el que se explica las distintas acciones que se pueden realizar a lo largo de la navegación del sitio web. Se dividirá en dos partes:

1. Manual de usuario de la aplicación de escraqueo
2. Manual de uso de la página web de deporte base

3.1. MANUAL DE USUARIO APLICACIÓN DE ESCRAPEO

En la página de inicio hay varias opciones para comenzar el escraqueo. Por defecto la primera opción que sale marcada es la del escraqueo de futnavarra, y dentro de este las categorías.



Figura 3.1.1 Página de inicio

Con la opción de futnavarra seleccionada se pueden escrapear varios ítems: categorías, equipos, tipos de clasificación y resultados y clasificaciones.



Figura 3.1.2 Página de inicio, opciones futnavarra

A parte de la opción de futnavarra hay otras dos: intermunicipal y fútbol txiki.

Figura 3.1.3 Página de inicio, opciones torneos o campeonatos

Si se escoge la opción de scrapear intermunicipal, no hay opciones, directamente escrapea todo lo necesario.

Figura 3.1.4 Página de inicio, opción torneo Intermunicipal

Si se escoge la opción de escrapear fútbol txiki la aplicación llega a una pantalla intermedia en la que nos da la opción de escoger un archivo de nuestro ordenador que debe ser un archivo Excel. Con la flecha se puede volver a la pantalla de inicio del escrapeo.



Figura 3.1.5 Pantalla carga Excel

Si el archivo escogido no es un Excel aparece un error y la aplicación se mantiene en la pantalla de subida del Excel.



Figura 3.1.6 Página error el archivo no es un Excel

Una vez escogida cualquier opción, la aplicación llega a una pantalla de carga mientras realiza el escraqueo.



Figura 3.1.7 Pantalla cargando

Al finalizar aparecerá una pantalla con el mensaje “el scrapeo se ha realizado correctamente” y una opción para volver al inicio.

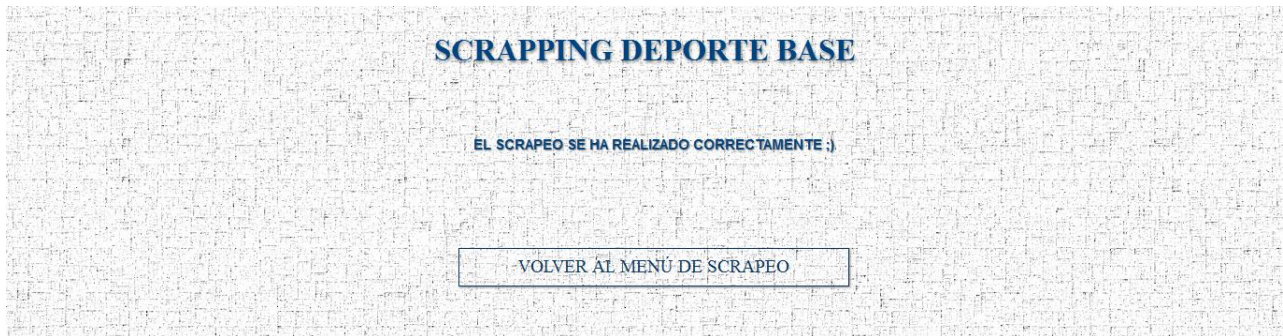


Figura 3.1.8 Página fin de escrapeo

Si se intentan realizar dos escrapeos al mismo tiempo, la aplicación llega a una pantalla con el mensaje “ya hay un escrapeo en curso, espere a que termine”, con una opción para volver al inicio.

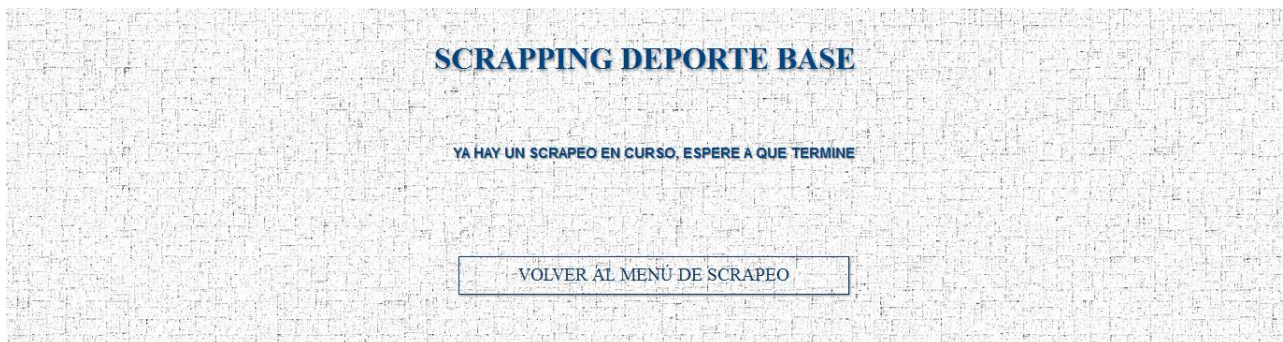


Figura 3.1.9 Pantalla escrapeo en curso

3.2. MANUAL DE USO DE LA WEB DE DEPORTE BASE

La pantalla de inicio consta de un menú y tres botones. Los botones llevan al mismo sitio que los ítems principales del menú, a una pantalla intermedia con todas las opciones que podemos visualizar: equipos (el listado de los equipos), clasificaciones (última tabla de clasificación guardada), partidos (resumen de los partidos jugados), y gráficas.



Figura 3.2.1 Página de inicio



Figura 3.2.2 Página intermedia de los campeonatos

Si se escoge la opción de equipos, aparece una pantalla con una tabla con el listado de equipos de una temporada, categoría, grupo y fase determinados. Por defecto aparece el primer listado almacenado en la base de datos. También existe la opción de filtrar esta tabla por temporada, categoría, fase y grupo para encontrar el listado de equipos que se desee⁴. Como extra se ha añadido un cuadro de texto “Nombre equipo” en el que por cada carácter introducido aparecerá en la tabla los equipos que contienen esa cadena⁵.



The screenshot shows the 'EQUIPOS INTERMUNICIPAL' screen. At the top, there are navigation tabs: INICIO, FUTNAVARRA, INTERMUNICIPAL (selected), and FÚTBOL TXIKI. Below the tabs is a title 'EQUIPOS INTERMUNICIPAL' with left and right arrow icons. Underneath is a 'FILTROS' section with the following filters: NOMBRE EQUIPO (empty), TEMPORADAS: 2015-16, CATEGORÍAS: 9 AÑOS, GRUPOS: A, and VUELTAS: 1. Below the filters is a table with 4 columns: NOMBRE, CATEGORIA, GRUPO, and VUELTA.

| NOMBRE | CATEGORIA | GRUPO | VUELTA |
|--------------------|-----------|-------|--------|
| Mendillorri b | 9 AÑOS | A | 1 |
| Avance Ezcabarte B | 9 AÑOS | A | 1 |
| Huerto D | 9 AÑOS | A | 1 |
| Sanduzelai A | 9 AÑOS | A | 1 |
| Jaso A | 9 AÑOS | A | 1 |
| Santa Catalina B | 9 AÑOS | A | 1 |

Figura 3.2.3 Pantalla lista equipos



The screenshot shows the 'EQUIPOS INTERMUNICIPAL' screen with the same navigation tabs and title. In the 'FILTROS' section, the NOMBRE EQUIPO filter is now set to 'MeN'. The table below shows the filtered results.

| NOMBRE | CATEGORIA | GRUPO | VUELTA |
|---------------|-----------|-------|--------|
| Mendillorri b | 9 AÑOS | A | 1 |
| Mendillorri A | 9 AÑOS | H | 1 |
| Mendillorri A | 9 AÑOS | F | 2 |
| Mendillorri B | 9 AÑOS | H | 2 |

Figura 3.2.4 Pantalla lista equipos filtrados por nombre

⁴ Estos filtros se repiten tanto en las opciones de visualizar clasificación como en las opciones de visualizar partidos del campeonato Intermunicipal y del campeonato de Fútbol txiki.

⁵ El filtro por nombre aparece siempre que se quiera visualizar los listados de equipos de cualquier campeonato.

En el caso de futnavarra, al tener una gran cantidad de datos, se ha decidido no almacenar todas las temporadas, solo la actual. Entonces los filtros cambian ya que no se puede filtrar por temporada. Sin embargo, diferencia entre dos juegos, sala y campo, y dentro de estos dos sexos, masculino y femenino. En resumen, en la opción de futnavarra los filtros son: juego, categoría, grupo y fase⁶.

| NOMBRE | CATEGORIA | GRUPO | FASE |
|---------------------------------|----------------------|-------|------|
| C.D.Tafatrans Vulcanizados Ruiz | Copa Cadete Femenino | 1 | 1 |
| C.D.Cantera | Copa Cadete Femenino | 1 | 1 |
| C.D.Orvina | Copa Cadete Femenino | 1 | 1 |
| C.D.Municipal De Buñuel | Copa Cadete Femenino | 1 | 1 |
| C.D.Aoiz | Copa Cadete Femenino | 1 | 1 |

Figura 3.2.5 Pantalla lista equipos futnavarra

Si la opción escogida es la de clasificaciones el funcionamiento es el mismo que en el caso de equipos. Aparece una tabla con la clasificación más reciente de la categoría, grupo y fase seleccionados.

| NOMBRE EQUIPO | JUGADOS | GANADOS | EMPATADOS | PERDIDOS | GOLES A FAVOR | GOLES EN CONTRA | GOLAVERAGE | PUNTOS |
|--------------------|---------|---------|-----------|----------|---------------|-----------------|------------|--------|
| Mendillorri b | 10 | 9 | 0 | 1 | 9 | 1 | 8 | 18 |
| Avance Ezcabarte B | 10 | 7 | 0 | 3 | 7 | 3 | 4 | 14 |
| Huerto D | 10 | 7 | 0 | 3 | 7 | 3 | 4 | 14 |
| Sanduzelai A | 10 | 3 | 0 | 7 | 3 | 7 | -4 | 6 |
| Jaso A | 10 | 2 | 0 | 8 | 2 | 8 | -6 | 4 |
| Santa Catalina B | 10 | 2 | 0 | 8 | 2 | 8 | -6 | 4 |

Figura 3.2.6 Pantalla clasificación

⁶ Estos filtros se repiten en las opciones de visualizar clasificaciones y partidos de futnavarra.

Si la opción escogida es la de partidos, aparecen todos los partidos jugados hasta la fecha de los equipos pertenecientes a la categoría, grupo y fase seleccionados. Esta opción tiene un extra, posibilita filtrar los partidos por fecha, para así poder visualizar el último jugado.

INICI

FUTNAVARRA

INTERMUNICIPAL

FÚTBOL TXIKI

←

PARTIDOS FÚTBOL TXIKI

→

FILTROS

TEMPORADAS:

2015-16

CATEGORÍAS:

6 AÑOS

GRUPOS:

A

FASES:

2

FECHAS

TEMPORADA:

Seleccione una temporada...

MES:

Seleccione un mes...

Día:

Seleccione un día...

BUSCAR

| FECHA | NOMBRE EQUIPO CASA | NOMBRE EQUIPO VISITANTE | RESULTADO | CIRCUNSTANCIA |
|---------------------|--------------------|-------------------------|-----------|---------------|
| 27 de Abril de 2016 | GAZTE BERRIAK K | ROCHAPEA A | 1 - 0 | |
| 27 de Abril de 2016 | MENDILLORRI I | IRABIA H | 1 - 0 | |
| 27 de Abril de 2016 | SAN CERNIN K | ARDOI E | 0 - 1 | |
| 2 de Mayo de 2016 | GAZTE BERRIAK K | ROCHAPEA A | 1 - 0 | |
| 2 de Mayo de 2016 | MENDILLORRI I | IRABIA H | 1 - 0 | |

Figura 3.2.7 Pantalla partidos

INICIO

FUTNAVARRA

INTERMUNICIPAL

FÚTBOL TXIKI

←

PARTIDOS FÚTBOL TXIKI

→

FILTROS

TEMPORADAS:

2015-16

CATEGORÍAS:

6 AÑOS

GRUPOS:

A

FASES:

2

FECHAS

TEMPORADA:

2015-16

MES:

Mayo

Día:

30

BUSCAR

| FECHA | NOMBRE EQUIPO CASA | NOMBRE EQUIPO VISITANTE | RESULTADO | CIRCUNSTANCIA |
|--------------------|--------------------|-------------------------|-----------|---------------|
| 30 de Mayo de 2016 | GAZTE BERRIAK K | ROCHAPEA A | 1 - 0 | |
| 30 de Mayo de 2016 | MENDILLORRI I | IRABIA H | 1 - 0 | |
| 30 de Mayo de 2016 | SAN CERNIN K | ARDOI E | 0 - 1 | |

Figura 3.2.8 Pantalla partidos filtrados por fecha

Si la opción escogida es la de gráficas, la aplicación llega a una aplicación intermedia para escoger entre tres gráficos. Para los tres campeonatos se pueden visualizar los gráficos de radar y de líneas paralelas. En el caso de los campeonatos Intermunicipal y Fútbol Txiki se puede visualizar un calendario de calor y en futnavarra un gráfico de barras. El contenido de estos gráficos se explica a continuación.



Figura 3.2.9 Pantalla intermedia gráficas Intermunicipal y fútbol txiki



Figura 3.2.10 Pantalla intermedia gráficas futnavarra

Si el gráfico escogido es el calendario de calor, se debe escoger una temporada y una fase. Entonces aparecerá un gráfico de calor que representa un resumen por jornadas de los resultados de cada equipo. Mediante colores se representa si el equipo en esa jornada a ganado, perdido, empatado o no ha obtenido resultado ya sea porque ha descansado o porque no consta. Además, al pasar el cursor del ratón por cada celda aparece un resumen del partido jugado en esa jornada por el equipo. Como extra, una vez aparece la gráfica, se puede escribir en el campo de texto “Nombre equipo” el nombre de tu equipo para buscarlo más fácilmente. También hay un icono de información que al pasar el cursor sobre él explica el proceso a seguir⁷.



Figura 3.2.11 Gráfico calendario de calor



Figura 3.2.12 Gráfico calendario de calor ayuda

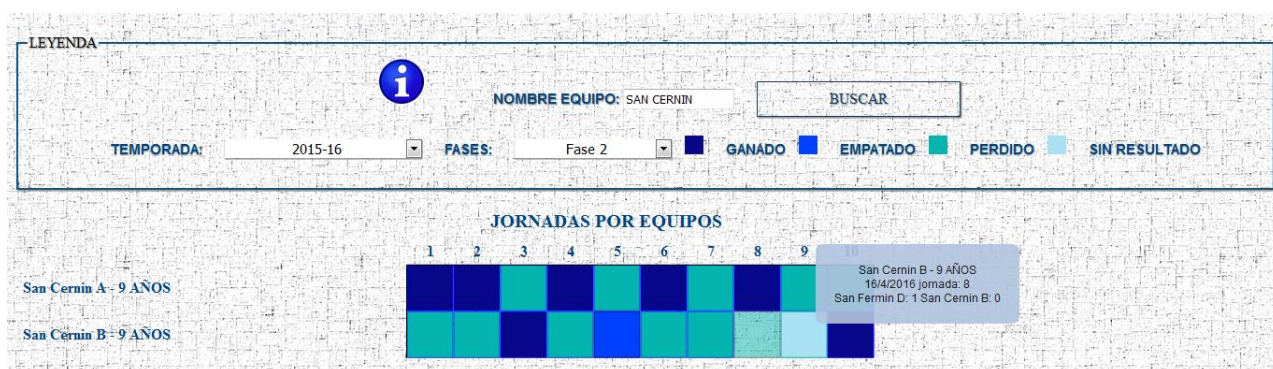


Figura 3.2.13 Gráfico calendario de calor filtrado por nombre

⁷ Este icono aparece en todas las gráficas.

Si el gráfico escogido es el de radar, hay que escoger dos equipos para poder compararlos. Esta comparación se realiza mediante los datos cuantitativos de la clasificación más reciente. Para que quede más claro a qué equipo pertenece cada área al pasar por encima con el ratón informa a que equipo pertenece.

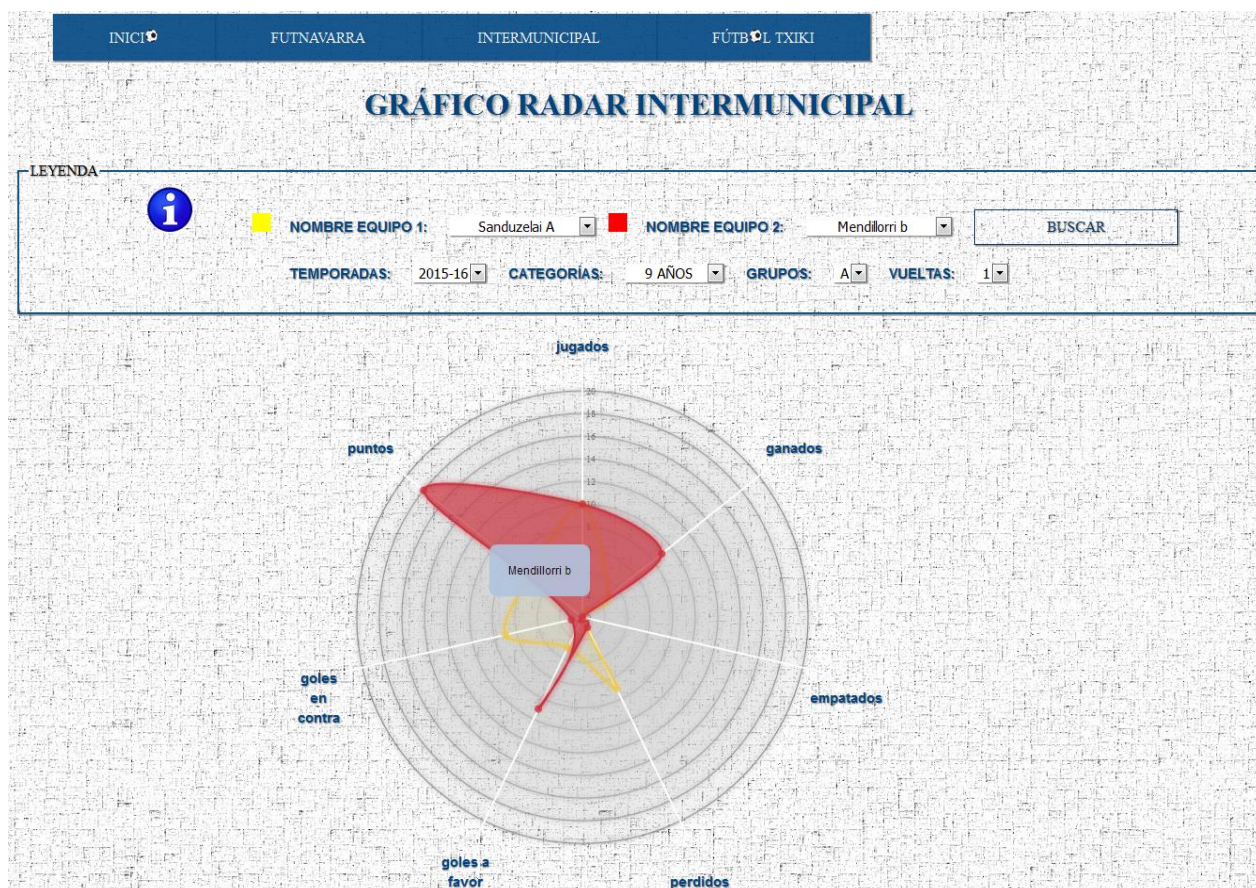


Figura 3.2.14 Gráfico radar

Si la opción escogida es el gráfico de líneas paralelas aparece un gráfico en el que se muestra la posición en la tabla de clasificación de cada equipo a través de las jornadas. Así podemos ver la evolución de los equipos a lo largo del campeonato. Si nos posicionamos encima de una línea o un nombre solo se marcan la línea y el nombre correspondientes.



Figura 3.2.15 Gráfico de líneas paralelas

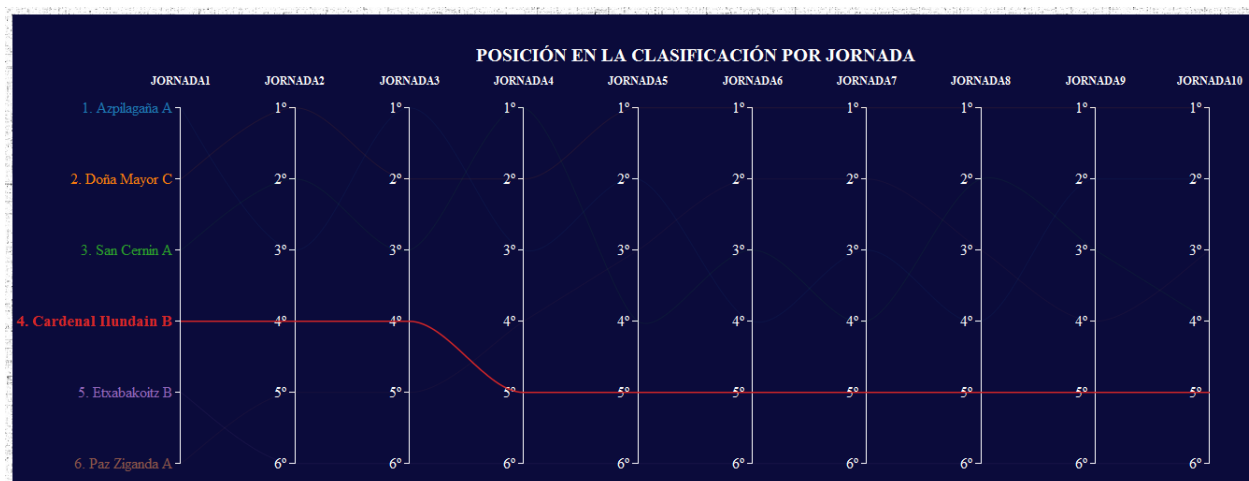


Figura 3.2.16 Gráfico de líneas paralelas equipo seleccionado

Por último, si el gráfico escogido es el gráfico de barras del campeonato futnavarra, nos aparece en que 10 categorías el equipo seleccionado es el mejor. Esto se cuantifica según los partidos ganados.

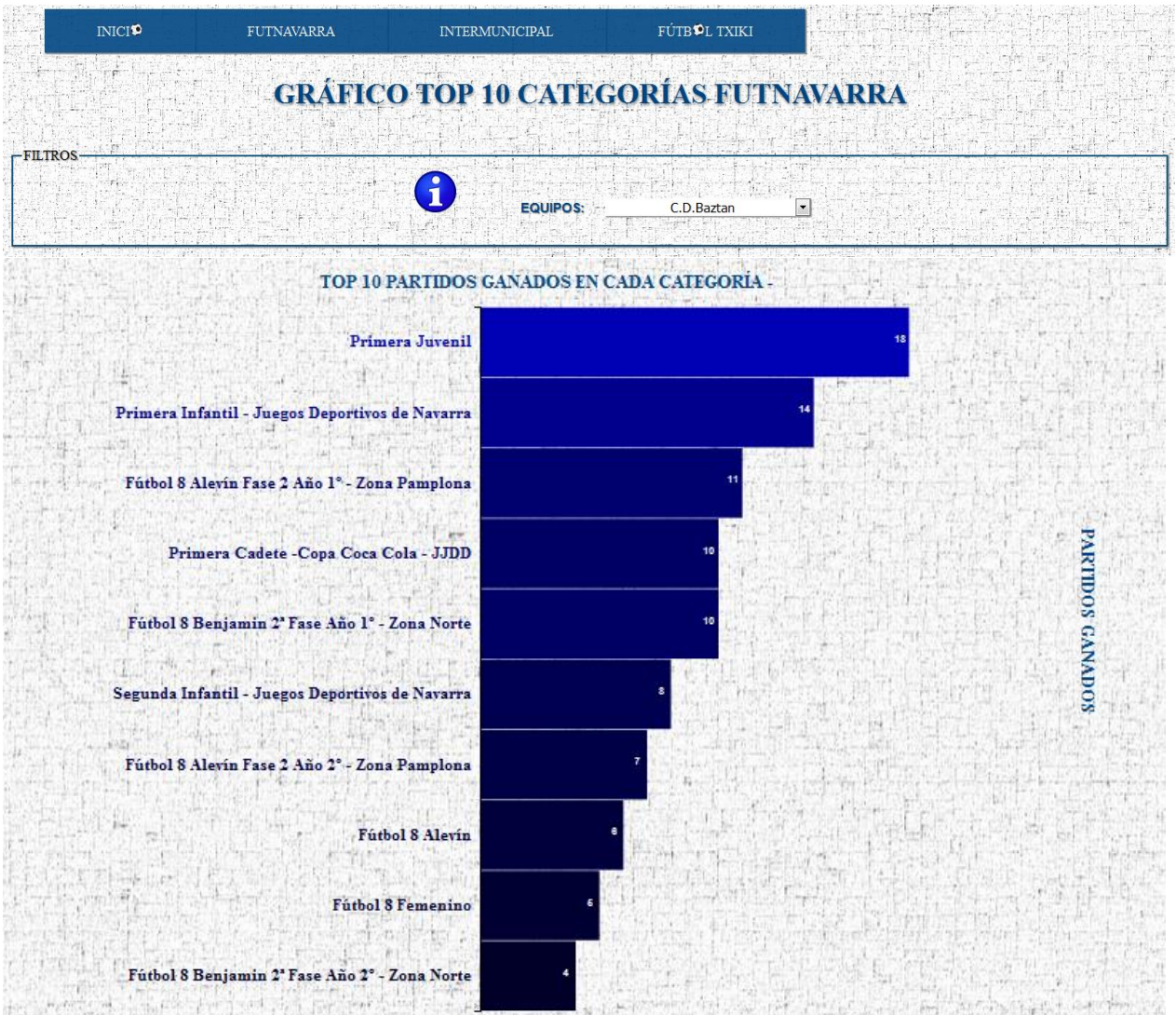


Figura 3.2.17 Gráfico de barras top 10 categorías por equipo futnavarra

CAPITULO IV DESPLIEGUE

4. DESPLIEGUE

El despliegue de la aplicación se ha realizado en el hosting web Webfaction. A continuación, se explica cómo se ha realizado este proceso. Hay que mencionar que este proceso es exclusivo de Webfaction, aunque puede ser similar en otros hostings.

Para desplegar la aplicación en el servidor se han seguido los siguientes pasos:

1. Creación de la base de datos vacía en phpMyAdmin
2. Creación de un proyecto en Webfaction. El proyecto se crea con un nombre cualquiera, en este caso se ha elegido proyectoana. En este paso hay que indicar con que framework se ha realizado la aplicación y en que versión.

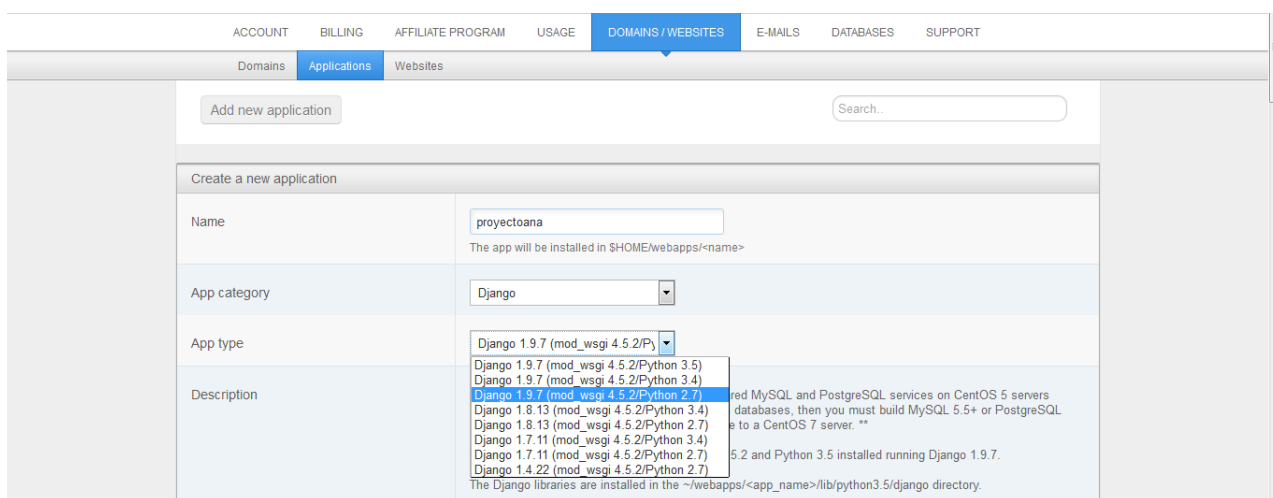


Figura 4.1 Creación de un proyecto en Webfaction

3. Una vez creado el proyecto se le asigna una ruta para acceder a él mediante el navegador.

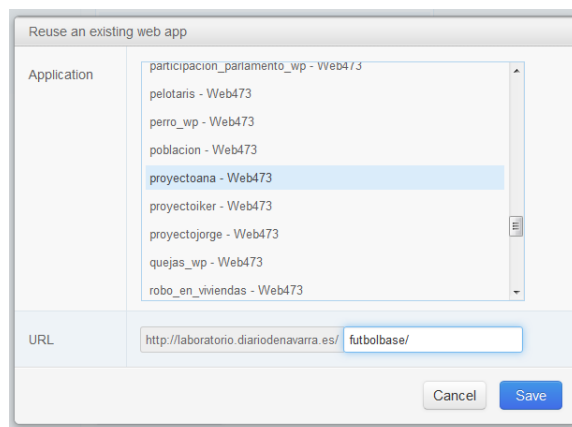
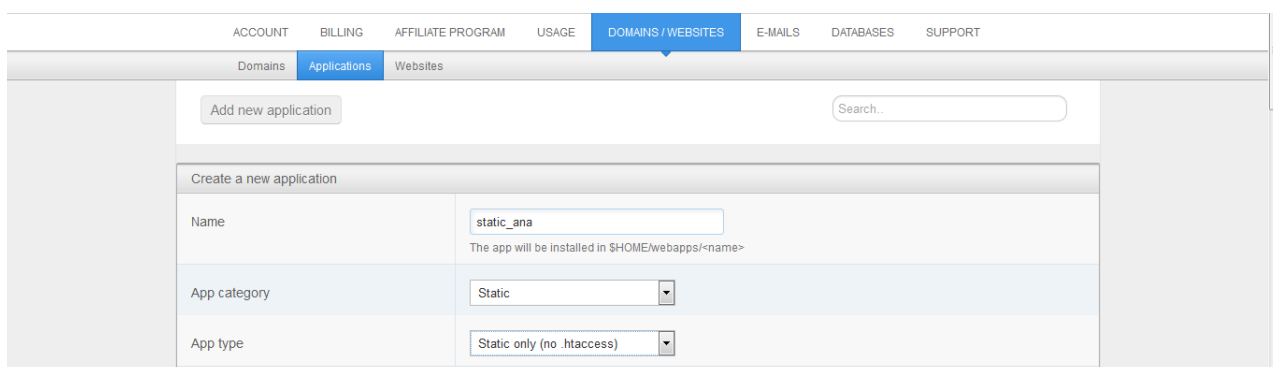


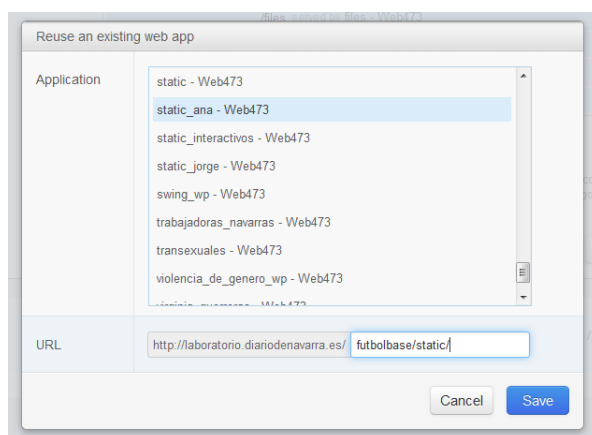
Figura 4.2 Asignación de la ruta al proyecto

- Al tratarse de un proyecto implementado con el framework Django hay que realizar de nuevo los pasos 2 y 3 para alojar los archivos estáticos que utiliza la web (imágenes, archivos css, archivos javascript).



The screenshot shows the 'DOMAINS / WEBSITES' section of a control panel. Under the 'Applications' tab, there is a 'Create a new application' form. The form includes a search bar, a 'Add new application' button, and three input fields: 'Name' (containing 'static_ana'), 'App category' (set to 'Static'), and 'App type' (set to 'Static only (no .htaccess)'). A note below the name field states: 'The app will be installed in \$HOME/webapps/<name>'.

Figura 4.3 Creación del proyecto que almacenará los archivos estáticos



The screenshot shows a 'Reuse an existing web app' dialog box. It contains a list of applications under the heading 'Application'. The list includes: 'static - Web473', 'static_ana - Web473' (which is highlighted), 'static_interactivos - Web473', 'static_jorge - Web473', 'swing_wp - Web473', 'trabajadoras_navarras - Web473', 'transexuales - Web473', and 'violencia_de_genero_wp - Web473'. Below the list, there is a 'URL' field with the value 'http://laboratorio.diariodenavarra.es/futbolbase/static/'. At the bottom right, there are 'Cancel' and 'Save' buttons.

Figura 4.4 Asignación de la ruta al proyecto static_ana

- Una vez creados ambos proyectos y asignadas sus rutas se procede a incluir los archivos con el código de programación en el proyecto. Hay que tener en cuenta que al crear el proyecto Webfaction ha creado la carpeta myproject donde irán alojados los archivos creados. Lo primero a hacer es cambiar este nombre por el asignado durante la fase de desarrollo, en este caso dbsite.
- Una vez cambiado el nombre se procede a copiar los archivos necesarios del proyecto (archivos HTML y archivos de Python). Esto se realiza mediante un cliente FTP, en este caso se ha utilizado Filezilla.

7. Una vez llegado este punto hay que configurar adecuadamente apache y Django:
 - a. Mediante una sesión ssh se realiza una conexión al proyecto creado.
 - b. A continuación, se debe realizar:
 - i. Cambiar la configuración de apache. Esto se refiere a cambiar el fichero httpd.conf para que la ruta de Python del WSGIDaemonProcess's apunte al proyecto creado, ya que apunta a myproject:

nano ./webapps/proyectoana/apache2/conf/httpd.conf

```
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule wsgi_module modules/mod_wsgi.so
LoadModule unixd_module modules/mod_unixd.so

LogFormat "%{X-Forwarded-For}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
CustomLog /home/user/logs/user/access_proyectoana.log combined
ErrorLog /home/user/logs/user/error_proyectoana.log

Listen 27033
KeepAlive Off
SetEnvIf X-Forwarded-SSL on HTTPS=1
ServerLimit 1
StartServers 1
MaxRequestWorkers 5
MinSpareThreads 1

WSGIDaemonProcess proyectoana python-
path=/home/user/webapps/proyectoana:/home/user/webapps/proyectoana/myproject:/home/user/webapps/proyectoana/lib/python2.7
WSGIProcessGroup proyectoana
WSGIRestrictEmbedded On
WSGILazyInitialization On
WSGIScriptAlias / /home/user/webapps/proyectoana/myproject/myproject/wsgi.py
```

Figura 4.5 Configuración de apache2 original

```
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule wsgi_module modules/mod_wsgi.so
LoadModule unixd_module modules/mod_unixd.so

LogFormat "%{X-Forwarded-For}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
CustomLog /home/user/logs/user/access_proyectoana.log combined
ErrorLog /home/user/logs/user/error_proyectoana.log

Listen 27033
KeepAlive Off
SetEnvIf X-Forwarded-SSL on HTTPS=1
ServerLimit 1
StartServers 1
MaxRequestWorkers 5
MinSpareThreads 1

WSGIDaemonProcess proyectoana python-
path=/home/user/webapps/proyectoana:/home/user/webapps/proyectoana/dbsite:/home/user/webapps/proyectoana/lib/python2.7
WSGIProcessGroup proyectoana
WSGIRestrictEmbedded On
WSGILazyInitialization On
WSGIScriptAlias / /home/user/webapps/proyectoana/dbsite/dbsite/wsgi.py
```

Figura 4.6 Configuración de apache2 tras el cambio

- ii. Configurar settings.py para que apunte al proyecto creado, añadir las líneas:
 ROOT_URLCONF = 'dbsite.urls'
 WSGI_APPLICATION = 'dbsite.wsgi.application'
- iii. Configurar settings.py para poder conectarse a las bases de datos creadas:

```
DATABASES = {  
    'nombre_bbdd_paraDjango': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'nombre_bbdd',  
        'USER': 'usuario',  
        'PASSWORD': 'contraseña',  
    },  
}
```

En este caso en ENGINE se ha escogida esta opción porque el sistema de gestión de base de datos que se utiliza es mysql, pero podría ser otro.

- iv. Configurar settings.py para poder leer los archivos estáticos, añadir las líneas:

```
STATIC_URL =  
'http://laboratorio.diariodenavarra.es/futbolbase/static'  
STATIC_ROOT = '/home/user/webapps/static_ana'
```

- v. Una vez hecho esto se reinicia el servidor apache:
 ./webapps/proyectoana/apache2/bin/restart

8. Y ya está todo listo para verlo en la web.

CAPITULO V ACCESIBILIDAD

5. ACCESIBILIDAD

En este apartado se hace un recorrido sobre las pautas de accesibilidad dictadas por la Web Content Accessibility Guidelines 2.0, en castellano Pautas de Accesibilidad para el Contenido Web, incidiendo en si la web de visualización de datos las cumple.

Primero es necesario explicar que es la accesibilidad, y que es la WCAG 2.0. La accesibilidad se define como la capacidad de acceso que tiene la web. Esto quiere decir que cualquier usuario puede usar la web independientemente de sus conocimientos técnicos o incluso de sus discapacidades. La WCAG 2.0 [4] es un documento que recoge una serie de pautas o recomendaciones para hacer los sitios web más accesibles. En concreto me he centrado en los dictados por la ONCE en su guía de referencia de accesibilidad web [5].

A continuación, se explican algunas de esas pautas y se comprueba que la aplicación de visualización de datos las cumple:

1. **Principio 1: Perceptible:** *“La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos”.*

En general la web no tiene un contenido difícil de entender ya que utiliza el formato básico de visualización de datos en el ámbito del deporte (tablas). El contenido más difícil de entender son las gráficas a las que se ha añadido un icono de información, que culturalmente todos los usuarios saben que significa, con instrucciones para su uso.

En cuanto al diseño, al elegir un fondo claro y como contenido el color azul, todo se distingue a la perfección. Como problema se podría mencionar el gráfico de calor que utiliza distintas tonalidades de azul para dar la información y quizá esto sea un problema para los usuarios con visibilidad reducida.

En cada tabla se ha implementado un efecto zoom al pasar por encima de cada elemento bajando el zoom del resto. Esto ayuda a las personas con visibilidad reducida a ver mejor el contenido.

2. **Principio 2: Operable:** *“Los componentes de la interfaz y la navegación deben ser operables”.*

Todos los elementos de la web son accesibles mediante teclado, lo que facilita la labor si no se posee un ratón. Además, cada vez que un elemento tiene el foco aparece seleccionado para saber dónde estamos.

El orden de las tabulaciones está ordenado según se disponen los elementos en la web para facilitar esta tarea.

Todas las páginas de la web están tituladas adecuadamente para que el usuario detecte en todo momento donde se encuentra.

Por último, se puede acceder al contenido de las páginas por varias vías:

1. Mediante un menú desplegable
2. Pulsando en los ítems del menú posicionados más arriba, que llevan a pantallas intermedias con las mismas opciones que el desplegable.

3. **Principio 3: Comprensible:** *“La información y el manejo de la interfaz de usuario debe ser comprensible”.*

El idioma de la web es solo y exclusivamente castellano ya que se trata de una web de deporte base navarro. Como posible mejora se podría añadir en un futuro el cambio de idioma a euskera ya que también es idioma local.

Es bastante predecible de utilizar, cada opción a la que se puede acceder tiene un nombre significativo con el que se deduce cual es el contenido.

El único error que puede dar la web es el caso de que no haya datos recogidos, en ese caso aparece en la pantalla el mensaje “NO HAY DATOS DISPONIBLES”, para informar al usuario de este hecho.

4. **Principio 4: Robusto:** *“El contenido debe ser suficientemente robusto para que pueda ser interpretado por una amplia variedad de agentes de usuario, incluyendo los productos de apoyo”.*

De momento la web solo está probada para pantallas de ordenador, pero está programado que en un futuro se pueda acceder desde cualquier dispositivo.

CAPITULO VI
PRUEBAS DE
USABILIDAD

6. PRUEBAS DE USABILIDAD

En este apartado se detallan las pruebas de usabilidad que se han llevado a cabo en la web y los resultados obtenidos.

En primer lugar, se ha utilizado un cuestionario SUS (System Usability Scale) [6] que mediante 10 preguntas mide el resultado obtenido sobre 100 puntos. Este cuestionario se encuentra en el [ANEXO II](#).

Previo a la realización del cuestionario se han planteado una serie de pruebas a los usuarios para que valoren la web objetivamente. Estas pruebas se encuentran en el [ANEXO III](#).

El número de usuarios utilizado para estas pruebas ha sido reducido ya que dada la baja complejidad de la web no se ha estimado oportuno realizársela a más. Entre estas personas se han seleccionado usuarios con ningún conocimiento informático y sin costumbre a utilizar el ordenador, usuarios habituados al uso de la web, pero sin conocimientos deportivos, y usuarios habituados al uso de la web y a visitar sitios web deportivos.

Tras realizar las pruebas a los usuarios y completar los cuestionarios se ha llegado a un resultado medio de 84 puntos, lo que denota que la web es considerablemente usable. Entre estos resultados el más alto ha sido de 90 puntos y el más bajo de 80 que no distan mucho entre sí.

Las preguntas mejor valoradas han sido las referidas a la facilidad de uso de la web y la facilidad de aprendizaje para utilizarla. Cabe destacar que un único usuario a reflejado que no utilizaría la web con frecuencia.

Tras la realización de esta prueba se puede concluir que el resultado es satisfactorio y que la web es usable para el 84% de los usuarios, aunque podría mejorar algunos aspectos. También se ha concluido que los usuarios sin conocimientos deportivos previos y sin costumbre a utilizar sitios web deportivos, tienen más dificultad en la interpretación de los gráficos de la web, pero con las explicaciones textuales introducidas en el sitio este hecho se subsana.

CAPITULO VII
PROBLEMAS Y
SOLUCIONES

7. PROBLEMAS ENCONTRADOS Y SOLUCIONES PROPUESTAS

Durante el desarrollo del proyecto se han sucedido una serie de problemas que han requerido una solución. En este apartado se describen los problemas y las soluciones que se han llevado a cabo para subsanarlos.

7.1 ESCRAPEO FUTNAVARRA

Durante el escraqueo de la página de la federación navarra de fútbol (www.futnavarra.es) surgieron varios problemas:

1. Problema: por lo general a la hora de realizar un escraqueo se buscan etiquetas significativas para acceder al contenido deseado, como, por ejemplo, el id de un elemento. Este sitio web está implementado de tal forma que no se ha usado el atributo id para identificar a los elementos lo que ha dificultado la tarea.

Solución: al no poder realizar el escraqueo buscando los atributos id se propuso buscarlos por las etiquetas de sus elementos según la información requerida (<select>, <table>, <div>, etc.), por lo general la información se encuentra en desplegables y en tablas.

2. Problema: este sitio web está construido y estructurado mediante tablas lo que complica la tarea de buscar la tabla concreta que tiene la información buscada.

Solución: para solucionar este problema se propusieron dos vías:

- a) buscar el número de tabla que tiene la información buscada o
- b) buscando palabras clave en el html (Clasificación, Jornada, Grupo, etc).

Cualquiera de las dos soluciones es válida, pero acarrearán un problema a futuro, si la web en un futuro cambia su implementación, estructura o palabras usadas la implementación realizada para el escraqueo será errónea.

7.2 LECTURA DE ARCHIVO EXCEL

La lectura del archivo Excel no ha resultado ningún problema actualmente. Es una lectura sencilla en la que se buscan una serie de palabras que nos indica que contenido estamos leyendo para poder almacenarlo adecuadamente. Originalmente no se planteó un formato concreto para el archivo Excel, lo que planteó un problema, no se pueden almacenar coherentemente datos que no se sabe dónde están. Para solucionar este hecho se realizó una plantilla estándar que es la que se debe rellenar para poder escrapear los datos. Esto plantea un problema a futuro, si la plantilla cambia o se sube un archivo que no cumpla la plantilla, el escraqueo fallará. En el [ANEXO IV](#) se puede ver la plantilla.

7.3 GRÁFICAS DE LA WEB DE VISUALIZACIÓN

Las gráficas implementadas en la web de visualización de datos se decidieron a posteriori de la implementación del escraqueo. Esto supuso un problema por los datos que se requería mostrar. Cuando se decidió qué se iba a almacenar en la base de datos uno de los requisitos era que para ahorrar espacio no se iba a almacenar la clasificación de cada jornada, si no la más reciente. Dos de los gráficos impuestos después requerían saber las clasificaciones de todas las jornadas para poder realizar comparativas de evolución de los equipos. Esto se solucionó adaptando el escraqueo a este hecho añadiendo el atributo jornada en la base de datos y almacenando las nuevas clasificaciones sin sobrescribir las ya almacenadas.

7.4 ERROR EN EL DESPLIEGUE

Durante el despliegue de la aplicación llegó un momento en el que no se podía acceder a la web porque el servidor dejó de procesar peticiones. Este se debió a que el servidor tiene un tope de procesos que puede ejecutar al mismo tiempo y el tope había llegado. Por este hecho se llegó a la conclusión de que el servidor no cumplía los requisitos que requería la aplicación y no se realizó el despliegue.

7.5 IMPLEMENTACIÓN INCOHERENTE FUTNAVARRA

Uno de los problemas que más ha complicado la tarea del escraqueo ha sido la incoherencia de la implementación de la web de la Federación Navarra de Fútbol. Las categorías por lo general juegan dos fases, una de septiembre a enero y otra de febrero a mayo. En ocasiones esto se refleja mediante un desplegable adicional para poder elegir la fase, otras veces, se crea una categoría nueva que incluye en el nombre Fase 2 o 2ª Fase o F2, etc. Esto ha obligado a que durante el escraqueo se compruebe el nombre de la categoría para saber si ya por sí sola pertenece a una fase.

CAPITULO VIII CONCLUSIONES

8. CONCLUSIONES Y MEJORAS FUTURAS

Tras la realización de este proyecto se pueden sacar 5 conclusiones:

1. Es muy importante realizar correctamente la implementación de las páginas web para que tengan más posibilidades.
2. Django es un framework que ayuda mucho a la implementación web limpia y sencilla.
3. El uso de d3js para la implementación de gráficos aporta muchas funcionalidades y efectos para realizarlos fácilmente y con un diseño bonito.
4. El scraping web, aunque aún no sea una técnica legal del todo, aporta mucho valor a un sitio web ya que, si se implementa correctamente, ahorra el tiempo de administración de datos.
5. Es muy importante la accesibilidad y la usabilidad web, puesto que, no todos los usuarios son iguales y no tienen las mismas capacidades, pero si el mismo derecho a navegar por la web.

Las posibles mejoras a futuro comienzan por el diseño de la aplicación. Se ha apostado por un diseño austero puesto que el fin de la web es mostrar datos y no requiere que sea bonita, sino que se vea clara la información.

Está pensada para cualquier deporte, no solo para fútbol, así que una mejora sería su ampliación a cualquier deporte que muestre en una web la información necesaria.

Está pensada para ser visitada desde cualquier dispositivo. Este hecho a día de hoy no se ha comprobado ya que el despliegue no se realizado, pero está pensado realizarlo en un futuro.

Por último, la realización de otros muchos gráficos con otro tipo de información que interese a los potenciales usuarios.

ANEXOS

ANEXO I: FRAMEWORK DJANGO

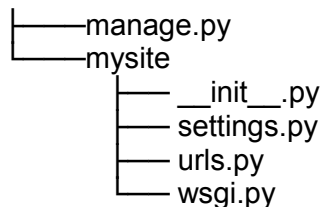
Para representar mejor el funcionamiento de Django, conviene explicar cuál es su estructura de archivos y para qué sirven estos. Primero se define que es un proyecto Django, seguido de cuál es la estructura de archivos y la explicación de algunos de esos archivos. A continuación, se detalla el contenido del fichero de configuración terminando con la explicación de la realización de las consultas a la base de datos.

A) Proyecto Django:

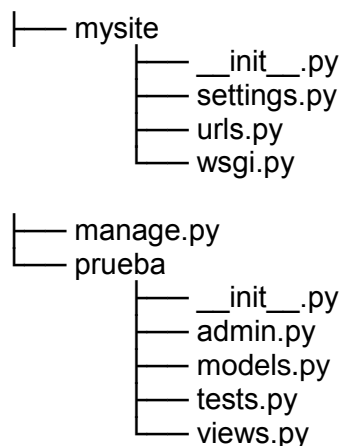
“Un proyecto es una colección de configuraciones para una instancia de Django, incluyendo configuración de base de datos, opciones específicas de Django y configuraciones específicas de aplicaciones.” [7]

B) Estructura de archivos:

1. Estructura de archivos de un proyecto → si se crea un proyecto llamado “mysite”, la estructura que crea Django es:



2. Estructura de archivos de una app dentro de un proyecto → si se crea una app llamada prueba dentro del proyecto mysite, la estructura que se crea es:



C) Archivos:

1. `__init__.py` → este archivo hace que Python trate el directorio del Proyecto como un paquete.
2. `manage.py` → utilidad de línea de comandos para poder interactuar con el proyecto de varias formas.
3. `settings.py` → archivo que almacena la configuración del proyecto.
4. `urls.py` → contiene la declaración de las urls del proyecto.

D) Fichero de configuración:

Hay que prestar especial atención a los siguientes ítems de configuración:

1. **DEBUG** → indica si la aplicación está en fase de desarrollo (true) o de producción (false). La diferencia es que cuando ocurre un error, si está a true, aparecerá una pantalla con todos los detalles del error incluyendo datos del servidor. Si está a false aparecerá una pantalla de error genérica.
2. **INSTALLED_APPS** → es la lista de las aplicaciones instaladas en el proyecto. Aquí debemos incluir todas las aplicaciones creadas.
3. **ROOT_URLCONF** → indica donde está el fichero urls.py en el que están las urls que usará la aplicación.
4. **DATABASES** → es el listado de bases de datos a las que se conectará el proyecto. Por cada base de datos hay que indicar como se llamará en el proyecto, en que motor de base de datos está alojada, el nombre de la base de datos, el usuario y la contraseña.

```
'inter': {  
    'ENGINE': 'django.db.backends.mysql',  
    'NAME': 'INTERMUNICIPAL',  
    'USER': 'user',  
    'PASSWORD': 'pass',  
},
```

Figura A1.1 Configuración de una base de datos en Django

5. **LANGUAGE_CODE** y **TIME_ZONE** → idioma que va a utilizar el proyecto y en qué zona horaria se encuentra.
6. **STATIC_URL** → donde se encuentran alojados los archivos estáticos.
7. **MEDIA_ROOT** → donde se alojarán los archivos subidos al servidor.

E) Consultas:

Django traduce las tablas de la base de datos a objetos definidos en el archivo models.py. Si por ejemplo en la base de datos existe una tabla llamada Categorías con los atributos id, nombre, fase y temporada en el archivo models.py se habrá creado:

```
class Categorías(models.Model):
    id = models.IntegerField(db_column='ID', primary_key=True) # Field name made lowercase.
    nombre = models.CharField(db_column='NOMBRE', max_length=50) # Field name made lowercase.
    fase = models.IntegerField(db_column='FASE') # Field name made lowercase.
    temporada = models.IntegerField(db_column='TEMPORADA') # Field name made lowercase.

    class Meta:
        managed = False
        db_table = 'CATEGORIAS'
```

Figura A1.2 Clase Categorías creada en el modelo de Django

Para realizar una consulta que nos devuelva todos los registros de la tabla Categorías, hay que hacer referencia a la clase Categorías, indicar que se quiere obtener los objetos, a que base de datos corresponde ese modelo e indicar que queremos todos:

```
todas = Categorías.objects.using('inter').All()
```

También se pueden utilizar filtros. Por ejemplo, obtener todas las categorías que pertenezcan a la fase 2:

```
categoríasFase2 = Categorías.objects.using('inter').filter(fase=2)
```

Además, existe la opción de poder ordenar los registros obtenidos. Por ejemplo, obtener todas las categorías pertenecientes a la fase 2 ordenadas por nombre.

```
categoríasFase2Ordenadas =
Categorías.objects.using('inter').filter(fase=2).orderby('nombre')
```

Esto es un breve resumen de las facilidades que aporta Django a la hora de obtener datos de una base de datos, existen otras muchas funcionalidades.

ANEXO II: CUESTIONARIO SUS

Debe responder a cada pregunta de 1 a 5, siendo 1 muy en desacuerdo y 5 muy de acuerdo.

1. Creo que usaría esta web con frecuencia.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

2. Creo que la web es innecesariamente compleja.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

3. Creo que la web es fácil de usar.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

4. Creo que es necesario el apoyo de un experto para navegar en el sitio web.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

5. Creo que las funcionalidades de la web están bien integradas.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

6. Creo que hay demasiada inconsistencia en la web.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

7. Creo que la mayoría de las personas aprenderían rápidamente a usar la web.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

8. Creo que la web es complicada de usar.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

9. Me he sentido seguro en el uso de la web.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

10. Es necesario aprender muchas cosas antes de usar la web.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

ANEXO III: PRUEBAS DE USABILIDAD

A continuación, se describen las pruebas a realizar por los usuarios antes de contestar las preguntas del formulario de usabilidad. Todos los usuarios realizarán todas las pruebas, iniciadas desde la página de home de la web. Previamente se dejará unos 5 minutos al usuario para que se familiarice con la aplicación.

Pruebas:

1. Dígame que equipos hay en el grupo C de la categoría 9 AÑOS del torneo Intermunicipal esta temporada en la fase 2. A continuación, vuelva a la página de inicio.
2. Busque el gráfico que visualiza la evolución de los equipos en las últimas 10 jornadas del torneo Intermunicipal. A continuación, dígame que equipo del grupo A de la categoría 9 AÑOS de esta temporada en la fase 2 ha sido el más estable. ¿Y el más fluctuante?
3. Acceda a los partidos del torneo de Futnavarra. Busque los partidos disputados por los equipos de fútbol sala masculino pertenecientes al grupo 2 de la categoría Benjamín F2 Año 2. Dígame los resultados de los partidos disputados el 23 de abril de 2016. A continuación, haga que solo se muestren los disputados en esa fecha.
4. Busque el gráfico que resume el resultado de cada equipo en cada jornada del campeonato intermunicipal. Dígame el resultado en cada jornada del equipo Lacturale Xota C de la categoría 9 AÑOS (si ha ganado, empatado, perdido o no hay resultado). A continuación, haga que en la gráfica solo se muestren los equipos que se llamen Lacturale.
5. Acceda a los equipos pertenecientes al torneo Fútbol Txiki. Dígame a que categoría, grupo y fase pertenece el equipo Huerto B esta temporada.
6. Busque el gráfico que muestra en que categorías destacan los equipos del torneo Futnavarra. Dígame en que categoría destaca el equipo C.D. Lourdes y cuantos partidos ha ganado en esta categoría. ¿Y el U.C.D. Burlades?
7. Busque el gráfico que hace comparativas entre dos equipos del torneo Intermunicipal. Compare los equipos Lacturale Xota E y Anaitasuna E del grupo B de la categoría 10-11 AÑOS de la fase 2 esta temporada. ¿Cuántos puntos tiene cada uno? En tu opinión ¿cuál es mejor y por qué?

ANEXO IV: PLANTILLA EXCEL

| | | | |
|--------------------------------|--|--|--|
| CATEGORÍA X - GRUPO X - X FASE | | | |
| | | | |
| | | | |
| | | | |

| | | | | | | | |
|---------------|------|------|------|------|------|------|--------|
| CLASIFICACIÓN | | | | | | | |
| EQUIPO | P.J. | P.G. | P.E. | P.P. | G.F. | G.C. | PUNTOS |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| | | | |
|--------------------------------|--|--|--|
| CATEGORÍA X - GRUPO X - X FASE | | | |
| | | | |
| | | | |
| | | | |

| | | | | | | | |
|---------------|------|------|------|------|------|------|--------|
| CLASIFICACIÓN | | | | | | | |
| EQUIPO | P.J. | P.G. | P.E. | P.P. | G.F. | G.C. | PUNTOS |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| | | | |
|--------------------------------|--|--|--|
| CATEGORÍA X - GRUPO X - X FASE | | | |
| | | | |
| | | | |
| | | | |

| | | | | | | | |
|---------------|------|------|------|------|------|------|--------|
| CLASIFICACIÓN | | | | | | | |
| EQUIPO | P.J. | P.G. | P.E. | P.P. | G.F. | G.C. | PUNTOS |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| | | | |
|--------------------------------|--|--|--|
| CATEGORÍA X - GRUPO X - X FASE | | | |
| | | | |
| | | | |
| | | | |

| | | | | | | | |
|---------------|------|------|------|------|------|------|--------|
| CLASIFICACIÓN | | | | | | | |
| EQUIPO | P.J. | P.G. | P.E. | P.P. | G.F. | G.C. | PUNTOS |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

BIBLIOGRAFÍA

1. "Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation", *Crummy.com*, 2016. [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.
2. M. Bostock, "D3.js - Data-Driven Documents", *D3js.org*, 2016. [Online]. Available: <https://d3js.org/>.
3. "W3Schools Online Web Tutorials", *W3schools.com*, 2016. [Online]. Available: <http://www.w3schools.com/>.
4. "Web Content Accessibility Guidelines (WCAG) 2.0", *W3.org*, 2016. [Online]. Available: <https://www.w3.org/TR/WCAG20/>.
5. *Guía de referencia: Accesibilidad de páginas web*, 1st ed. ONCE-CIDAT, 2013, pp. 36-39. [Online]. Available: <ftp://ftp.once.es/pub/utt/biblioteca/Accesibilidad/webs/AccesibilidadWeb2013.pdf>.
6. A. Affairs, "System Usability Scale (SUS)", *Usability.gov*, 2016. [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.
7. A. Holovaty and J. Kaplan-Moss. "Empezando" en *El libro de Django*, J. Dunck, 2008, pág. 9.
8. A. Holovaty and J. Kaplan-Moss, *El libro de Django*. Jeremy Dunck, 2008.
9. D. Mordecki, *Miro y Entiendo*. 2012.
10. S. Krug, *No me hagas pensar*. Madrid: Prentice Hall, 2001.